SCRAPE: Tính ngẫu nhiên có thể mở rộng được chứng thực bởi các thực thể công khai

Ignacio Cascudo[1*] và Bernardo David[23**]

[1]Đại học Aalborg, Đan Mạch
[2]Đại học Aarhus, Đan Mạch
[3]IOHK, Hong Kong

Tóm lược. Các Beacon ngẫu nhiên (một cơ chế được sử dụng để tạo ra các giá trị ngẫu nhiên có thể dự đoán và không thể được chi phối bởi bất kỳ bên nào) thống nhất có đầu ra có thể được chứng thực công khai là không thiên vị được yêu cầu trong một số giao thức mật mã. Một cách tiếp cận phổ biến để xây dựng các Beacon như vậy là yêu cầu một số bên chạy giao thức tung đồng xu (Coin Tossing) với phân phối đầu ra được đảm bảo (để kẻ tấn công không thể dễ dàng ngăn các bên trung thực nhận được tính ngẫu nhiên, do đó tạm dừng các giao thức dựa vào nó). Tuy nhiên, các công trình hiện tại phải đối mặt với các vấn đề nghiêm trọng về khả năng mở rộng do chi phí tính toán và truyền thông cao. Chúng tôi trình bày một giao thức tung đồng xu cho đa số trung thực, cho phép bất kỳ thực thể nào xác minh rằng đầu ra được tạo một cách trung thực bằng cách quan sát thông tin có sẵn công khai (ngay cả sau khi quá trình thực thi hoàn tất), đồng thời đạt được cả phân phối đầu ra được đảm bảo và khả năng mở rộng. Khối xây dựng chính trong quá trình xây dựng của chúng tôi là lược đồ Chia sẻ bí mật có thể kiểm chứng công khai đầu tiên cho các cấu trúc truy cập ngưỡng chỉ yêu cầu phép tính $O(n)$. Các lược đồ trước đây yêu cầu các phép tính $O(nt)$ (trong đó t là ngưỡng) từ mỗi bên liên quan, khiến chúng không phù hợp để tạo tính ngẫu nhiên phân tán có thể mở rộng, yêu cầu $t = n/2$ và do đó, các phép tính $O(n^2)$.

1 Giới thiệu

Vấn đề có được một nguồn ngẫu nhiên đáng tin cậy đã được nghiên cứu từ những ngày đầu tiên của mật mã học. Trong khi các bên riêng lẻ có thể chọn nguồn ngẫu nhiên tin cậy có sẵn một cách cục bộ, thì người ta đã chứng minh nguồn ngẫu nhiên cục bộ có thể bị thay đổi [BLN16, DPSW16] và nhiều ứng dụng yêu cầu một nguồn ngẫu nhiên công khai chung được đảm bảo không thiên vị bởi một kẻ tấn công có khả năng. Sự cần thiết này đã truyền cảm hứng cho công việc quan trọng về Tung đồng xu của Blum [Blu81], cho phép 2 hoặc nhiều bên tạo ra một đầu ra được đảm bảo là ngẫu nhiên thống nhất miễn là ít nhất một trong các bên trung thực (và với điều kiện là giao thức kết thúc).

* Ignacio Cascudo ghi nhận sự hỗ trợ từ Hội đồng Nghiên cứu Độc lập Đan Mạch, cấp số. DFF-4002-00367.

** Dự án này đã nhận được tài trợ từ Hội đồng nghiên cứu châu Âu (ERC) theo chương trình nghiên cứu và đổi mới Horizon 2020 của Liên minh châu Âu (thỏa thuận tài trợ số 669255).

Khái niệm về *Beacon ngẫu nhiên* công khai phát hành định kỳ các giá trị ngẫu nhiên không thiên vị và không thể đoán trước mới được đề xuất bởi Rabin [Rab83] trong bối cảnh ký kết hợp đồng và đã tìm thấy một số ứng dụng khác như giao thức bỏ phiếu [Adi08], tạo tham số công khai cho sơ đồ mã hóa [BDF+15, LW15], thông điệp tức thời bảo vệ quyền riêng tư [WCGFJ12, vdHLZZ15] và duyệt web ẩn danh [DMS04, GRFJ14]. Gần đây hơn, các ứng dụng Blockchain [Nak08, GKL15] như hợp đồng thông minh [KMS+16, B+14], Sharding [CDE+16] và các giao thức đồng thuận dựa trên Proof-of-Stake [KKR+16] đã làm tăng nhu cầu về các nguồn ngẫu nhiên [BCG15].

Khái niệm về Beacon ngẫu nhiên của Rabin rất phù hợp với các ứng dụng trên nhưng việc triển khai được đề xuất trong [Rab83] phụ thuộc vào một bên thứ ba đáng tin cậy. Mục tiêu của bài nghiên cứu này là xây dựng một Beacon ngẫu nhiên phân tán đảm bảo phân phối đầu ra và tính ngẫu nhiên được phân phối đồng đều cho các bên sử dụng Beacon miễn là phần lớn các bên này trung thực. Ngoài ra, trong nhiều ứng dụng đã nói ở trên, các bên không nhất thiết phải tham gia vào việc tạo ngẫu nhiên nhưng muốn kiểm tra việc thực thi giao thức phải có khả năng chứng thực *hậu nghiệm* rằng nguồn ngẫu nhiên là đáng tin cậy và không thiên vị. Do đó, chúng tôi nhắm đến việc xây dựng một Beacon ngẫu nhiên có thể kiểm chứng công khai và không chỉ là một giao thức tạo ra tính ngẫu nhiên cho các bên tham gia tích cực vào việc thực hiện nó.

1.1 Công việc liên quan

Một giải pháp tự nhiên để có được các cảnh báo về tính ngẫu nhiên bao gồm sử dụng giao thức tung đồng xu do Blum [Blu81] đề xuất với các thông điệp của nó được đăng lên bảng thông báo công khai để xác minh sau này (hoặc truyền phát giữa các bên). Tuy nhiên, người ta biết rằng trong trường hợp một nửa hoặc nhiều bên bị hỏng, kẻ tấn công có thể làm sai lệch đầu ra của giao thức hoặc thậm chí ngăn cản các bên trung thực nhận được bất kỳ đầu ra nào bằng cách hủy bỏ việc thực thi giao thức tại một điểm nhất định [Cle86]. Giả sử phần lớn người chơi là trung thực, có thể đảm bảo phân phối đầu ra [RBO89] thông qua ngưỡng chia sẻ bí mật có thể kiểm chứng (VSS - Verifiable Secret Sharing) [CGMA85] với điều kiện là có sẵn một kênh truyền phát. Về cơ bản, với điều kiện phần lớn n bên là trung thực, mỗi bên có thể bí mật chia sẻ thông tin đầu vào của mình thành n phần chia sẻ (Share) sao cho $n/2$ là đủ để tái tạo bí mật, gửi một phần chia sẻ cho mỗi bên liên quan trước khi bắt đầu giao thức tung đồng xu. Mặc dù kẻ tấn công không thể khôi phục bất kỳ đầu vào nào (vì nó có tối đa $n/2 - 1$ phần chia sẻ của mỗi đầu vào), tập thể các bên trung thực biết ít nhất $n/2$ phần chia sẻ mà họ có thể sử dụng để tái tạo đầu vào của các bên hủy bỏ và sau đó kết thúc giao thức.

Mặc dù giao thức tung đồng xu với phân phối đầu ra được đảm bảo (GOD - Guaranteed Output Delivery) với đa số trung thực dựa trên VSS cung cấp nguồn ngẫu nhiên đáng tin cậy, cách tiếp cận này vẫn có hai vấn đề chính: 1. Hầu hết các kế hoạch VSS đều yêu cầu sự tương tác giữa người phân phối (Dealer) và các bên khác cản trở khả năng mở rộng và 2. Chỉ những bên tham gia tích cực vào giao thức mới có thể xác minh rằng nó đã được thực thi chính xác. Mặc dù VSS không tương tác [Fel87] giải quyết vấn đề tương tác, nhưng nó không cho phép việc thực thi giao thức được xác minh độc lập bởi các thực thể không tham gia tích cực. Một cách tự nhiên để cho phép bất kỳ thực thể nào xác minh đầu ra do các giao thức đó tạo ra thực sự được tạo ra một cách

trung thực là thay thế VSS truyền thống bằng các kế hoạch chia sẻ bí mật có thể kiểm chứng công khai (PVSS - Publicly Verifiable Secret Sharing) [Sta96], cho phép bất kỳ ai xác minh tính hợp lệ của các chia sẻ và bí mật được xây dựng lại thông qua thông tin có thể được cung cấp công khai mà không yêu cầu tương tác trực tiếp giữa bất kỳ bên nào. Các biến thể của phương pháp này đã được đề xuất trong [KKR$^+$16, SJK$^+$16].

Trong khi [KKR$^+$16] khởi tạo giao thức tung đồng xu GOD kiểu [RBO89] đơn giản yêu cầu giao tiếp giữa tất cả các bên (thông qua sổ cái công khai), [SJK$^+$16] giảm độ phức tạp của giao tiếp bằng cách chia các bên thành các ủy ban điều hành giao thức trong nội bộ với đầu ra có thể kiểm chứng công khai. Sau đó, một *máy trạm* chỉ liên lạc với *lãnh đạo* của mỗi ủy ban (thay vì nói chuyện với tất cả các bên) để tổng hợp các kết quả đầu ra này để có được tính ngẫu nhiên có thể kiểm chứng công khai. Tuy nhiên, trong khi cách tiếp cận ban đầu của [KKR$^+$16] đạt được sự an toàn khi giả định có đa số trung thực (có nghĩa là kẻ tấn công làm hỏng ít hơn một nửa số bên), thì cách tiếp cận hiệu quả về giao tiếp của [SJK$^+$16] chỉ đạt được sự an toàn trước một kẻ tấn công làm hỏng ít hơn một phần ba của tất cả các bên. Hơn nữa, với điều kiện là đa số trung thực, giao thức của [KKR$^+$16] đảm bảo rằng tất cả các bên đều nhận được đầu ra bất kể bên nào bị hỏng, trong khi ở các giao thức của [SJK$^+$16], ngay cả khi chỉ có máy trạm bị hỏng, nó có thể hủy bỏ và ngăn tất cả các bên khác nhận được sự ngẫu nhiên.

Mặc dù tung đồng xu với GOD được xây dựng thông qua PVSS có khả năng đạt được khả năng mở rộng và khả năng kiểm chứng công khai, các công trình PVSS hiện tại [Sta96, FO98, Sch99, BT99, RV05, HV09, Jha11, JVSN14] phải chịu chi phí tính toán cao. Nói chung, các bên được yêu cầu mỗi bên tính toán các phép tính $O(nt)$ để xác minh n phần chia sẻ bí mật với ngưỡng t, chuyển thành các phép tính $O(n^2)$ kể từ $t = n/2$ trong ứng dụng Beacon ngẫu nhiên của chúng tôi[1]. Chi phí tính toán này phát sinh do ý tưởng chính đằng sau các sơ đồ này là cam kết các hệ số của đa thức được sử dụng cho Chia sẻ bí mật Shamir [Sha79] và mã hóa các phần chia sẻ, sau đó sử dụng các cam kết đối với các hệ số để tính toán độc lập các cam kết đối với các phần chia sẻ được chứng minh là không kiến thức để tương ứng với các chia sẻ được mã hóa. Cách tiếp cận này ban đầu được đưa ra trong [Sch99], sử dụng phương pháp phỏng đoán Fiat-Shamir (và do đó là mô hình Oracle ngẫu nhiên) để thu được bằng chứng không kiến thức không tương tác cần thiết. Sau đó, các biến thể của giao thức này trong mô hình đơn giản đã được đề xuất trong [RV05, JVSN14], thay thế bằng chứng không kiến thức bằng kiểm tra dựa trên Mã hóa Paillier [Pai99] và trong [HV09, Jha11], đề xuất ghép nối dựa trên phương pháp kiểm tra tính hợp lệ của phần chia sẻ.

Các phương pháp khác để xây dựng các Beacon ngẫu nhiên công cộng đã được xem xét trong [BCG15, BDF$^+$15, LW15, BLMR14, BGM16]. Khả năng kiểm chứng công khai (hoặc khả năng kiểm toán) trong bối cảnh giao thức tính toán đa bên chung đã được xem xét trước đây trong [BDO14,SV15].

[1]Trên thực tế, [Jha11] cung cấp một giải pháp thay thế trong đó chỉ yêu cầu các phép tính $O(n)$ và một số cặp không đối để xác minh nhưng các cặp $O(n)$ được yêu cầu để thiết lập và các phép tính $O(n^2)$ trong nhóm mục tiêu của bản đồ kép (đắt hơn so với các phép tính khác được thực hiện trong các nhóm nguồn) được yêu cầu để tái tạo.

1.2 Đóng góp của chúng tôi

Chúng tôi giới thiệu SCRAPE, một giao thức triển khai Beacon ngẫu nhiên có thể kiểm chứng công khai với đa số trung thực thông qua giao thức tung đồng xu phân phối đầu ra được bảo đảm dựa trên PVSS. Kết quả chính của chúng tôi nằm ở cốt lõi của SCRAPE: sơ đồ PVSS ngưỡng *đầu tiên* chỉ yêu cầu một số lượng phép tính *tuyến tính* để chia sẻ, xác minh và tái tạo, trong khi các sơ đồ trước đó chỉ đạt được độ phức tạp bậc hai. Lược đồ PVSS này có thể được khởi tạo cả theo giả định Decisional Diffie Hellman (DDH) trong Mô hình Oracle ngẫu nhiên (ROM - Random Oracle Model) và trong *mô hình đơn giản* theo giả định Decisional Bilinear Square (DBS) [HV09]. Trong khi cải thiện độ phức tạp tính toán của các sơ đồ trước đó, sơ đồ PVSS của chúng tôi vẫn duy trì chi phí liên lạc thấp tương tự, làm cho nó phù hợp với các ứng dụng có số lượng lớn người dùng. Chúng tôi nhận xét rằng các kế hoạch PVSS mới của chúng tôi cũng có thể được sử dụng để cải thiện hiệu suất của [SJK+16].

Mô hình: Như trong các nghiên cứu trước đây [BDO14], chúng tôi giả định rằng các bên có thể sử dụng "bảng thông báo công khai" để công bố thông tin sẽ được sử dụng để xác minh sau. Trên thực tế, trong các ứng dụng mà chúng tôi quan tâm, có sẵn một *sổ cái* nơi các thông điệp có thể được đăng để xác minh sau, vì bản thân giao thức Bitcoin Backbone thực hiện một cơ chế như vậy (tức là sổ cái phân tán được phân tích trong [GKL15]). Tuy nhiên, các giao thức của chúng tôi tương thích với bất kỳ sổ cái công khai nào, không chỉ với [GKL15]. Khả năng mở rộng cũng là một mối quan tâm vì hàng trăm nghìn người dùng thường tham gia vào việc tạo ngẫu nhiên, giúp việc đăng thông điệp trong sổ cái có thể truy cập công khai trở nên đơn giản hơn thay vì yêu cầu tất cả các bên liên lạc với nhau.

Công nghệ của chúng tôi: Chúng tôi cải tiến sơ đồ PVSS của Schoenmakers [Sch99] và các biến thể của nó (yêu cầu các phép tính $O(nt)$ để xác minh n phần chia sẻ) bằng cách thiết kế một quy trình xác minh phần chia sẻ chỉ yêu cầu các phép tính $O(n)$ (hoặc ghép nối). Quy trình của chúng tôi khám phá thực tế rằng việc chia sẻ bí mật với Chia sẻ bí mật Shamir [Sha79] tương đương với việc mã hóa bí mật (cộng với tính ngẫu nhiên) bằng mã sửa lỗi Reed Solomon, một thực tế lần đầu tiên được quan sát bởi McEliece và cộng sự trong [MS81]. Vì các phần chia sẻ từ Chia sẻ bí mật Shamir tạo thành một Codeword của mã Code Reed Solomon, nên việc tính toán tích bên trong của một Vectơ chia sẻ với một Codeword từ mã kép tương ứng sẽ cho kết quả 0 nếu các phần chia sẻ được tính toán chính xác. Trong [Sch99], người phân phối trong sơ đồ của chúng ta chia sẻ bí mật bằng cách sử dụng Chia sẻ bí mật Shamir, mã hóa các chia sẻ $s_1,\ldots,s_n$ trong văn bản mã hoá có dạng $h^{sk_i\, s_i}$ (trong đó h^{sk_i} là khóa công khai và sk_i là khóa bí mật) nhưng cũng cam kết chia sẻ tất cả bằng cách tính toán $v_i = g^{s_i}$, trong đó g, h là hai trình tạo được chọn độc lập của một nhóm mà vấn đề DLOG được coi là khó. Người phân phối cũng cung cấp bằng chứng rằng phần chia sẻ trong văn bản mã hoá giống như phần chia sẻ trong các cam kết. Để xác minh tính hợp lệ của các chia sẻ, bất kỳ ai cũng có thể lấy mẫu từ mã ngẫu nhiên $\boldsymbol{c}^{\perp} = (\boldsymbol{c}_1^{\perp}, \ldots, \boldsymbol{c}_n^{\perp})$ của mã kép của mã Reed Solomon tương ứng với trường hợp chia sẻ Bí mật Shamir đã được sử dụng, tính tích

bên trong của $\boldsymbol{c}^{\perp}$ với các Vectơ chia sẻ trong phép tính của g (bằng các tính toán $\prod_i v_i^{\boldsymbol{c}_i^{\perp}} = g^{\sum_i s_i \boldsymbol{c}_i^{\perp}}$) và kiểm tra xem nó có bằng $g^0 = 1$ không. Nếu phần chia không hợp lệ, thì việc kiểm tra này có xác suất lớn là không thành công. Để chứng minh phần chia sẻ trong văn bản mã hoá và trong các cam kết là như nhau, người phân phối có thể sử dụng bằng chứng không kiến thức không tương tác (NIZK - Non-Interactive Zero-Knowledge) được xây dựng bằng cách sử dụng kinh nghiệm Fiat-Shamir như trong [Sch99] (dẫn đến việc xây dựng trong ROM theo giả định DDH) hoặc yêu cầu các bên thực hiện kiểm tra dựa trên ghép nối như trong [HV09] (dẫn đến một cấu trúc trong mô hình đơn giản theo giả định DBS).

Hiệu quả cụ thể: Trong cấu trúc dựa trên DDH trong ROM, người phân phối được yêu cầu tính toán 4*n* phép tính trong giai đoạn chia sẻ, trong khi xác minh và tái tạo tương ứng yêu cầu 4*n* phép tính và 5*t* + 3 (với điều kiện là tất cả *n* phần chia sẻ đều được xác minh nhưng chỉ *t* phần chia sẻ được sử dụng trong tái tạo). Trong cấu trúc dựa trên DBS trong mô hình đơn giản, người phân phối được yêu cầu tính toán 2*n* phép tính trong giai đoạn chia sẻ, trong khi xác minh yêu cầu 2*n* cặp và tái tạo yêu cầu 2*n* cặp và *t* + 1 phép tính (với điều kiện là *n* phần chia sẻ được giải mã được xác minh nhưng chỉ *t* phần chia sẻ được sử dụng trong quá trình tái tạo). Các kết quả trước đó [Sch99, HV09] yêu cầu khoảng *nt* phép tính bổ sung trong giai đoạn xác minh, dẫn đến $n^2/2$ phép tính bổ sung trong ứng dụng Beacon ngẫu nhiên, yêu cầu $t = n/2$. Trong xây dựng mô hình Oracle ngẫu nhiên, dữ liệu NIZK bổ sung là cần thiết, lên tới tổng số 2*n* phần tử nhóm và *n* +1 phần tử vòng do người phân phối công bố. Khi xây dựng ở mô hình đơn giản, người phân phối tiết kiệm dữ liệu NIZK và chỉ đăng 2*n* phần tử nhóm, trong khi yêu cầu tính toán tốn kém hơn (tức là ghép nối).

1.3 Nội dung tiếp theo

Trong Phần 2, chúng tôi giới thiệu các ký hiệu và định nghĩa sẽ được sử dụng xuyên suốt bài nghiên cứu. Trong Phần 3, chúng tôi trình bày giao thức PVSS của chúng tôi dựa trên giả định DDH trong ROM. Trong Phần 4, chúng tôi giới thiệu giao thức PVSS ở chế độ đơn giản dựa trên giả định DBS. Trong Phần 5, chúng tôi xây dựng một Beacon ngẫu nhiên dựa trên các giao thức PVSS của chúng tôi. Trong Phần 6, chúng tôi phân tích độ phức tạp cụ thể và hiệu suất của các giao thức của chúng tôi và trình bày các điểm chuẩn dựa trên triển khai nguyên mẫu. Cuối cùng, trong Phần 7, chúng tôi kết luận với các hướng dẫn cho công việc trong tương lai.

2 Sơ bộ

Trong phần này, chúng tôi thiết lập ký hiệu và giới thiệu các định nghĩa sẽ được sử dụng trong suốt bài nghiên cứu. Chúng tôi ký hiệu lấy mẫu thống nhất một phần tử ngẫu nhiên x từ một tập hợp hữu hạn D bởi $x \leftarrow D$. Chúng tôi ký hiệu các Vectơ là $x = (x_1,\ldots,x_n)$. Chúng tôi ký hiệu tích trong của hai Vectơ $\boldsymbol{x}, \boldsymbol{y}$ là $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \sum_{1 \le i \le n} x_i \cdot y_i$. Vì lợi ích của ký hiệu, số nguyên n sẽ luôn được coi là số chẵn, do đó $n/2$ là một số nguyên.

Trong bài nghiên cứu này q sẽ luôn biểu thị một số nguyên tố. Chúng tôi ký hiệu $\mathbb{Z}_q$ là vành các số nguyên Modulo q và $\mathbb{G}$ là nhóm nhân hữu hạn cấp q. Vì q là số nguyên tố, $\mathbb{Z}_q$ là trường hữu hạn và $\mathbb{G}$ là nhóm tuần hoàn trong đó mọi phần tử $g \neq 1$ đều là một trình tạo. Chúng tôi ký hiệu $\mathbb{Z}_q[x]$ là vành đa thức một biến với hệ số $\mathbb{Z}_q$. Chúng tôi ký hiệu $log_g e$ là Logarit rời rạc của một phần tử $e \in \mathbb{G}$ với trình tạo $g \in \mathbb{G}$.

2.1 Lý thuyết mã hóa

Chúng tôi xác định mã $[n,k,d]$ C là mã sửa lỗi tuyến tính trên $\mathbb{Z}_q$ có độ dài n, kích thước k và khoảng cách nhỏ nhất d. Mã kép của nó $C^{\perp}$ là không gian Vectơ bao gồm tất cả các Vectơ $\boldsymbol{c} \in \mathbb{Z}^n_q$ sao cho $\langle \boldsymbol{c}, \boldsymbol{c^{\perp}} \rangle = 0$ với mọi $\boldsymbol{c}$ thuộc C. Mã kép $C^{\perp}$ của mã $[n,k,d]$ C là mã $[n, n-k, d^{\perp}]$ (với một vài $d^{\perp}$). Trong công việc này, chúng tôi sẽ sử dụng đại số tuyến tính cơ bản sau đây.

Bổ đề 1. *Nếu $\boldsymbol{v} \in \mathbb{Z}^n_q \setminus C$, và $\boldsymbol{c}^{\perp}$ được chọn ngẫu nhiên như nhau trong $C^{\perp}$ thì xác suất để $\langle v, c^{\perp} \rangle = 0$ chính xác là $1/q$.*

Chứng minh. Theo tính chất tuyến tính, $\boldsymbol{c}^{\perp} \in C^{\perp}$ là trực giao với $\boldsymbol{v}$ nếu nó cũng trực giao với mọi Vectơ trong mã D kéo dài bởi $\boldsymbol{v}$ và C, nghĩa là, khi và chỉ khi $\boldsymbol{c}^{\perp} \in D^{\perp}$. Vì $\boldsymbol{v} \notin C$, nên kích thước của D là $k+1$ và do đó không gian $D^{\perp}$ có kích thước $n-k-1$. Do đó, nếu $\boldsymbol{c}^{\perp}$ được chọn ngẫu nhiên đều trong $C^{\perp}$ thì xác suất để $\langle v, c^{\perp} \rangle = 0$ là

$$\frac{\#(D^{\perp})}{\#(C^{\perp})} = \frac{q^{n-k-1}}{q^{n-k}} = \frac{1}{q}.$$

Hơn nữa, trong công việc này, chúng tôi sẽ luôn theo giả định $n < q$ và chúng tôi sẽ sử dụng mã Reed-Solomon C có dạng sau

$$C = \{(p(1), p(2), ..., p(n)) : p(x) \in \mathbb{Z}_q[x], \deg p(x) \leq k-1\}$$

trong đó $p(x)$ dao động trên tất cả các đa thức trong $\mathbb{Z}_q[x]$ bậc nhiều nhất là $k-1$. Đây là mã $[n, k, n-k+1]$. $C^{\perp}$ kép của nó là một mã $[n, n-k, k+1]$, có thể được định nghĩa như sau

$$C^{\perp} = \{(v_1 f(1), v_2 f(2), ..., v_n f(n)) : f(x) \in \mathbb{Z}_q[x], \deg f(x) \leq n-k-1\}$$

cho các hệ số $v_i = \prod_{j=1, j \neq i}^{n} \frac{1}{i-j}$.

2.2 Chia Sẻ Bí Mật Shamir (Shamir Secret Sharing)

Lược đồ chia sẻ bí mật ngưỡng (n, t) cho phép *người phân phối D* chia bí mật s thành n phần chia sẻ $\boldsymbol{S} = (s_1, ..., s_n)$ được phân phối giữa n bên $P_1, ..., P_n$ sao cho có thể tái tạo lại bí mật đã cho t của các phần chia sẻ nhưng không có thông tin nào được tiết lộ nếu ít phần chia sẻ hơn được biết đến. Chúng tôi coi $\boldsymbol{S}$ là *Vectơ chia sẻ* của lược đồ chia sẻ bí mật. Lược đồ chia sẻ bí mật ngưỡng đầu tiên được giới thiệu bởi Shamir trong [Sha79]. Để tách một bí mật $s \in \mathbb{Z}_q$, người phân phối lấy mẫu $t-1$ hệ số ngẫu nhiên $c_1, ..., c_{t-1} \leftarrow \mathbb{Z}_q$ và xây dựng một đa thức $p(x) = s + c_1 x + c_2 x^2 + \cdots + c_{t-1} x^{t-1}$. Các phần chia sẻ được tính là $s_i = p(i)$ với $1 \leq i \leq n$. Một bên sở hữu t phần chia sẻ có thể sử dụng phép

nội suy Lagrange để khôi phục đa thức $p(x)$ và do đó thu được s. Mặt khác, một bên biết ít hơn t phần chia sẻ không có thông tin về bí mật. McEliece và cộng sự lần đầu tiên quan sát thấy rằng việc chia bí mật thành n phần chia sẻ với Chia sẻ Bí mật Shamir tương đương với việc mã hóa thông điệp $(x, c_1,..., c_{t-1})$ dưới mã Reed Solomon $[n, t, n-t+1]$, ngụ ý rằng Vectơ chia sẻ $\boldsymbol{S}$ là một Codeword của mã Reed Solomon như vậy.

2.3 Giả định

Một trong những cấu trúc của chúng tôi đã được chứng minh trong Mô hình Oracle ngẫu nhiên [BR93], trong đó giả định rằng các bên được cấp quyền truy cập vào một hàm $H(x)$ nhận đầu vào có kích thước bất kỳ và trả về các đầu ra ngẫu nhiên thống nhất duy nhất có kích thước cố định (trả về cùng một đầu ra mỗi khi đầu vào giống nhau). Một hàm như vậy có thể được khởi tạo trong thực tế bằng hàm Hash mật mã. Trong mô hình này, chúng tôi chứng minh tính bảo mật của các giao thức của mình theo giả định DDH, giả định rằng g, g^{α}, g^{β} khiến kẻ tấn công PPT khó có thể phân biệt giữa $g^{\alpha\beta}$ với g^{r}, trong đó g là trình tạo của nhóm $\mathbb{G}$ gồm lựa chọn q và $\alpha, \beta, r \leftarrow \mathbb{Z}_q$.

Nhóm song tuyến tính và giả định Decisional Bilinear Square. Như trong các tài liệu trước đây [HV09], chúng tôi đã chọn trình bày sơ đồ của mình trên một nhóm song tuyến tính đối xứng (ví dụ: Loại I trong thuật ngữ của [GPS06]). Tuy nhiên, cấu trúc của chúng tôi cũng có thể dễ dàng chuyển đổi thành các nhóm song tuyến tính không đối xứng (ví dụ: Loại III theo thuật ngữ của [GPS06]) [AHO16], đối với các đường cong thân thiện với cặp tuyến tính hiện đại [BN06] dành cho các thuật toán hiệu quả hơn để tính toán các cặp [AKL+11] đã được biết đến.

Định nghĩa 1. ***Nhóm song tuyến tính.*** *Một nhóm song tuyến tính được mô tả bởi một bộ* $\Lambda := (q, \mathbb{G}, \mathbb{G}_T, e)$ *trong đó* $\mathbb{G}$ *và* $\mathbb{G}_T$ *là các nhóm có thứ tự nguyên tố q và e là ánh xạ song tuyến tính* $\mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ *với các thuộc tính sau:*

- ***Song tuyến tính*** $e(g^{\alpha}, g^{\beta}) = e(g, g)^{\alpha\beta}$ *với mọi* $g \in \mathbb{G}$ *và* $\alpha, \beta \in \mathbb{Z}_q$.
- ***Không suy biến*** $e(g, g) \neq 1$ *trừ khi* $g = 1$.
- ***Hiệu quả*** *Tồn tại các thuật toán hiệu quả để tính toán các phép toán nhóm trong* $\mathbb{G}$, $\mathbb{G}_T$ *và để đánh giá* $e(x, y)$ *với* $x, y \in \mathbb{G}$.

Chúng tôi chứng minh giao thức dựa trên ghép nối của chúng tôi an toàn theo giả định Decisional Bilinear Square (DBS) [HV09] đã được trình bày trong bài nghiên cứu đó tương đương với giả định Thương số song tuyến tính quyết định [LV08] và liên quan đến giả định Song tuyến tính quyết định Diffie Hellman.

Giả định 1. ***Decisional Bilinear Square (DBS) [HV09].*** *Cho* $\Lambda := (q, \mathbb{G}, \mathbb{G}_T, e)$ *là một nhóm song tuyến tính. Đối với trình tạo* $g \in \mathbb{G}$, *các giá trị ngẫu nhiên* $\mu, v, s \leftarrow \mathbb{Z}_q$ *và cho* $u = g^{\mu}$ *và* $v = g^{v}$, *các phân phối xác suất sau không thể phân biệt bằng tính toán:* $D_0 = (g, u, v, T_0 = e(u, u)^{v})$ *và* $D_1 = (g, u, v, T_1 = e(u, u)^{s})$.

Mô hình kẻ tấn công. Chúng tôi chứng minh tính bảo mật của các giao thức của mình trong cài đặt độc lập chống lại các kẻ tấn công nguy hiểm, những người có thể đi chệch khỏi giao thức theo bất kỳ cách nào. Chúng tôi xem xét các kẻ tấn công tĩnh, những người phải chọn các bên để làm hỏng trước khi bắt đầu thực thi giao thức.

Sổ cái công khai và Kênh truyền phát. Chúng tôi giả định rằng các bên có quyền truy cập vào sổ cái công khai với *Tính sống động*, nghĩa là kẻ tấn công không thể ngăn cản các bên trung thực bổ sung thông tin và đồng ý về nó, và *Tính bền vững*, tức là thông tin không thể bị sửa đổi hoặc xóa sau đó. Một sổ cái như vậy có thể được triển khai bằng giao thức Bitcoin Backbone với giả định đa số trung thực, chữ ký số và Oracle ngẫu nhiên [GKL15]. Tuy nhiên, các giao thức của chúng tôi không dựa vào thuộc tính nào dành riêng cho sổ cái của [GKL15], nghĩa là các tài liệu của chúng tôi cũng có thể được khởi tạo trên sổ cái công khai trong mô hình đơn giản. Lưu ý rằng quyền truy cập vào một giao thức quảng bá thường được giả định trong các giao thức nhiều bên dành cho đa số trung thực [RBO89] và có thể đạt được hiệu quả tương tự của việc quảng bá một thông điệp bằng cách ghi nó vào sổ cái. Chúng tôi cũng nhận xét rằng tính khả dụng của *bảng thông báo công khai* đã được giả định trong các nghiên cứu trước đây về khả năng kiểm chứng công khai đối với các giao thức nhiều bên [BDO14, SV15].

2.4 Chia sẻ bí mật có thể kiểm chứng công khai (PVSS)

Chúng tôi áp dụng mô hình chung cho các lược đồ PVSS của [Sch99] và các định nghĩa bảo mật của [RV05, HV09] (với một số khác biệt mà chúng tôi nhận xét bên dưới). Chúng tôi xem xét một tập hợp n bên $P = \{P_1,...,P_n\}$ và một người phân phối D, người chia sẻ bí mật giữa tất cả các bên trong P. Chúng tôi sẽ xây dựng các lược đồ cho các cấu trúc truy cập ngưỡng (n, t), nghĩa là bí mật được chia thành n phần chia sẻ sao cho biết tối đa t - 1 phần chia sẻ không tiết lộ thông tin nhưng một tập hợp t phần chia sẻ cho phép tái tạo bí mật. Ngoài ra, bất kỳ trình xác minh bên ngoài V nào cũng có thể kiểm tra xem D có hành động trung thực hay không mà không cần biết bất kỳ thông tin nào về các phần chia sẻ hoặc bí mật. Một giao thức PVSS có bốn giai đoạn dưới đây:

- **Thiết lập.** Người phân phối D tạo và xuất bản các tham số của sơ đồ. Mỗi bên P_i công bố khóa công khai pk_i và giữ lại khóa bí mật tương ứng sk_i.
- **Phân phối.** Người phân phối tạo các phần chia sẻ $s_1,...,s_n$ cho bí mật s, mã hóa chia sẻ s_i bằng khóa công khai pk_i cho $i = 1,...,n$ và xuất bản các mã hóa $\hat{s}_i$, cùng với một bằng chứng $PROOF_D$ rằng đây thực sự là những mã hóa của việc chia sẻ hợp lệ một bí mật nào đó.
- **Xác minh.** Trong giai đoạn này, bất kỳ V bên ngoài nào (không nhất thiết phải là người tham gia giao thức) có thể xác minh mà không cần tương tác, với tất cả thông tin công khai cho đến thời điểm này rằng các giá trị $\hat{s}_i$ là mã hóa của việc chia sẻ bí mật hợp lệ.
- **Tái tạo.** Giai đoạn này được chia thành hai.
  Giải mã các phần chia sẻ: Giai đoạn này có thể được thực hiện bởi bất kỳ tập hợp Q nào của t hoặc nhiều bên. Mỗi bên P_i trong Q giải mã phần chia sẻ s_i từ văn bản

mã hoá $\hat{s}_i$ bằng cách sử dụng khóa bí mật sk_i và xuất bản s_i cùng với bằng chứng không kiến thức (không tương tác) $PROOF_i$ rằng giá trị này đúng là mã hoá của $\hat{s}_i$.
Chia sẻ tổng hợp: Bất kỳ trình xác minh bên ngoài *V* nào (không nhất thiết phải là người tham gia giao thức) đều có thể thực hiện giai đoạn này. Đầu tiên *V* kiểm tra xem các bằng chứng $PROOF_i$ có đúng không. Nếu việc kiểm tra vượt qua cho ít hơn *t* bên trong Q thì *V* hủy bỏ; mặt khác, *V* áp dụng thủ tục tái tạo cho tập hợp s_i các phần chia sẻ tương ứng với các bên P_i đã vượt qua kiểm tra.

Một chương trình PVSS phải cung cấp ba đảm bảo an toàn: Tính chính xác, Tính xác thực và Bảo mật IND1. Các thuộc tính này được định nghĩa dưới đây:

- **Tính chính xác.** Nếu người phân phối và tất cả người chơi trong Q đều trung thực, thì tất cả các kiểm tra xác minh trong giai đoạn xác minh và tái tạo đều đạt và bí mật có thể được tái tạo từ thông tin do người chơi trong Q công bố trong giai đoạn tái tạo.
- **Tính xác thực.** Nếu vượt qua việc kiểm tra trong bước xác minh thì khả năng cao là các giá trị $\hat{s}_i$ là mã hóa của việc chia sẻ hợp lệ của một bí mật. Hơn nữa, nếu quá trình kiểm tra trong giai đoạn Tái tạo được thông qua thì các giá trị được truyền đạt s_i đúng là phần chia sẻ bí mật được phân phối bởi người phân phối.
- **Bảo mật IND1 (IND1-Secrecy).** Trước giai đoạn tái tạo, thông tin công khai cùng với các khóa bí mật sk_i của bất kỳ nhóm tối đa t - 1 người chơi nào không cung cấp thông tin về bí mật. Về mặt hình thức, điều này được phát biểu như trong định nghĩa dựa trên tính không thể phân biệt được điều chỉnh từ [RV05,HV09] sau đây:

Định nghĩa 2. ***Tính không thể phân biệt của các bí mật (Bảo mật IND1).*** *Chúng tôi nói rằng PVSS là Bảo mật IND1 nếu đối với bất kỳ thời gian đa thức nào, kẻ tấn công* A_{Priv} *làm hỏng tối đa* $t-1$ *bên,* A_{Priv} *có lợi thế không đáng kể trong trò chơi tiếp theo đấu với người thách thức.*

1. *Người thách thức thực hiện giai đoạn Thiết lập của PVSS với tư cách là người phân phối và gửi tất cả thông tin công khai đến* A_{Priv}*. Hơn nữa, nó tạo ra các khóa bí mật và khóa công khai cho tất cả các bên không bị hỏng và gửi các khóa chung tương ứng tới* A_{Priv}*.*
2. A_{Priv} *tạo các khóa bí mật cho các bên bị hỏng và gửi các khóa công khai tương ứng cho người thách thức.*
3. *Người thách thức chọn ngẫu nhiên các giá trị* x_0 *và* x_1 *trong các bí mật. Hơn nữa, nó chọn* $b \leftarrow \{0, 1\}$ *một cách ngẫu nhiên. Nó chạy giai đoạn Phân phối của giao thức với* x_0 *là bí mật. Nó gửi* A_{Priv} *tất cả thông tin công khai được tạo trong giai đoạn đó, cùng với* x_b*.*
4. A_{Priv} *đưa ra dự đoán* $b' \in \{0, 1\}$.

Lợi thế của A_{Priv} *được định nghĩa là* $|\Pr[\, b = b' \,] - 1/2|$.

Bảo mật IND1 là định nghĩa được sử dụng trong [RV05,HV09], ngoại trừ thực tế là chúng tôi không áp đặt bất kỳ yêu cầu bảo mật nào sau giai đoạn Tái tạo. Sự khác biệt bắt nguồn từ thực tế là trong [RV05,HV09], yêu cầu không ai biết được bí mật ngoại trừ các bên tương tác trong quá trình tái tạo, trong khi trong ứng dụng Beacon ngẫu nhiên của chúng tôi, bí mật phải được tái tạo và xuất bản công khai. Chúng tôi nhận xét rằng sơ đồ của chúng tôi có thể đạt được cả định nghĩa thoải mái theo yêu cầu của ứng dụng Beacon ngẫu nhiên và đảm bảo bí mật mạnh mẽ hơn của [RV05,HV09] (thông qua việc sử dụng các kênh riêng tư giữa các bên hoặc thông qua kỹ thuật [HV09] yêu cầu thêm dữ liệu để được ghi vào sổ cái). Các kế hoạch của chúng tôi có thể đạt được khái niệm bí mật mạnh hơn được chính thức hóa là Bảo mật IND2 trong [RV05, HV09], cho phép kẻ tấn công lựa chọn bí mật tùy ý. Điều này được thực hiện bằng cách chuyển đổi hộp đen sang các giao thức cho phép chia sẻ các bí mật tùy ý thay vì các bí mật ngẫu nhiên bằng cách sử dụng bí mật được chia sẻ ngẫu nhiên làm "khoá dùng một lần (OTP)" để mã hóa bí mật tùy ý được chứng minh chính thức trong [RV05, HV09].

2.5 Cam kết

Các lược đồ cam kết [Blu81] là một mã hóa nguyên thủy cơ bản có chức năng như một hộp ký gửi kỹ thuật số an toàn. Về cơ bản, người gửi cam kết gửi thông điệp m bằng cách đặt nó vào trong hộp, khóa hộp và gửi hộp cho người nhận. Sau đó, người gửi có thể mở cam kết bằng cách đưa cho người nhận chìa khóa hộp, tiết lộ m. Lưu ý rằng người gửi không thể thay đổi thông điệp sau khi đưa hộp bị khóa cho người nhận (đây là thuộc tính *ràng buộc*) trong khi người nhận không thể tìm hiểu thông điệp trước khi nhận được khóa (đây là thuộc tính *ẩn*). Đối với các định nghĩa và cấu trúc chính thức của lược đồ cam kết với nhiều đảm bảo an toàn khác nhau và hiệu quả rất tốt, chúng tôi giới thiệu độc giả tới tài liệu sau cho các mô hình độc lập [Nao91] và Khả năng kết hợp chung [CDD+16]. Chúng tôi định nghĩa một cú pháp chung cho các cam kết như sau:

- **Com** (m, r) nhận đầu vào là thông điệp m và độ ngẫu nhiên r, xuất ra cam kết **Com** đối với thông điệp m.
- **Open** (m, r) nhận đầu vào là thông điệp m và độ ngẫu nhiên r, xuất thông tin mở cần thiết để kiểm tra xem cam kết **Com** tương ứng có hợp lệ đối với m và r không.

2.6 Bằng chứng không kiến thức (ZKP) về kiến thức Logarit rời rạc

Trong quá trình xây dựng dựa trên giả định DDH trong mô hình Oracle ngẫu nhiên, chúng tôi sẽ cần một bằng chứng không kiến thức của kiến thức về một giá trị $\alpha \in \mathbb{Z}_q$ sao cho $x = g^\alpha$ và $y = h^\alpha$ với g, x, h, y cho trước. Chúng tôi biểu thị bằng chứng này bằng *DLEQ* (g, x, h, y). Chaum và Pedersen đã xây dựng một giao thức Sigma để thực hiện bằng chứng này trong [CP93], giao thức của họ hoạt động như sau:

1. Trình chứng minh tính toán $a_1 = g^w$ và $a_2 = h^w$ trong đó $w \leftarrow \mathbb{Z}_q$ và gửi a_1, a_2 cho trình xác minh.

2. Trình xác minh gửi một thách thức $e \leftarrow \mathbb{Z}_q$ cho trình chứng minh.
3. Trình chứng minh gửi phản hồi $z = w - \alpha e$ cho trình xác minh.
4. Trình xác minh kiểm tra xem $a_1 = g^z x^e$ và $a_2 = h^z y^e$ và chấp nhận bằng chứng nếu điều này đúng.

Bằng chứng này có các đặc tính của tính đầy đủ, tính hợp lý và không kiến thức. Trong các bằng chứng của mình, chúng tôi sẽ đề cập cụ thể đến thuộc tính tính hợp lý, nghĩa là trình chứng minh không thể thuyết phục trình xác minh về một tuyên bố giả ngoại trừ *lỗi về tính hợp lý* không đáng kể. Lưu ý rằng giao thức Sigma này có thể được chuyển đổi thành bằng chứng không kiến thức không tương tác về kiến thức của α trong mô hình Oracle ngẫu nhiên thông qua phỏng đoán Fiat-Shamir [FS87, PS96]. Như trong [Sch99], chúng tôi cần tính song song chứng minh này cho n cặp giá trị riêng biệt (x_1,y_1), ..., (x_n, y_n). Trong trường hợp này, một thử thách duy nhất e được trình chứng minh tính là $e = H(x_1,y_1,...,x_n,y_n,a_{1,1},a_{2,1},...,a_{1,n},a_{2,n})$, trong đó các giá trị $a_{1,i}$,$a_{2,i}$ được tính theo x_i,y_i như mô tả ở trên và $H(\cdot)$ là một dự đoán ngẫu nhiên (tất nhiên có thể được thay thế bởi một hàm Hash mật mã). Sau đó, bằng chứng bao gồm thử thách e cùng với các câu trả lời z_i được tính toán theo từng x_i,y_i. Trình xác minh có thể kiểm tra bằng chứng bằng cách tính toán $a'_{1,i} = g^{z_i} x_i^e$ và $a'_{2,i} = h^{z_i} y_i^e$, và xác minh rằng $H(x_1, y_1,..., x_n, y_n, a'_{1,1}, a'_{2,i},..., a'_{1,n}, a'_{2,n}) = e$.

3 PVSS dựa trên giả định DDH trong ROM

Trong phần này, chúng tôi xây dựng giao thức PVSS an toàn theo giả định DDH trong Mô hình Oracle ngẫu nhiên. Chúng tôi tiếp cận chung tương tự như cách tiếp cận của Schoenmakers [Sch99] nhưng khác biệt đáng kể trong quy trình được sử dụng để xác minh phần chia sẻ, đại diện cho chi phí chính trong sơ đồ của Schoenmakers. Trong giai đoạn thiết lập, mỗi bên P_i được yêu cầu đăng ký trong sổ cái (hoặc truyền phát) khóa công khai pk_i có dạng $pk_i = h^{sk_i}$, trong đó h là trình tạo của nhóm $\mathbb{G}_q$ có thứ tự p và $sk_i \leftarrow \mathbb{Z}_q$ là khóa bí mật được mỗi bên lưu trữ. Trong giai đoạn Phân phối, người phân phối bắt đầu bằng cách chia sẻ bí mật $s \leftarrow \mathbb{Z}_q$ với Chia sẻ bí mật Shamir và mã hóa các phần chia sẻ $s_1,...,s_n$ dưới các khóa công khai đã đăng ký của các bên bằng cách tính toán $\hat{s}_i = pk_i^{k_i}$ để chia sẻ bí mật ngẫu nhiên của h^s. Nhưng thay vì cam kết với các hệ số của đa thức được sử dụng cho Chia sẻ bí mật Shamir, chúng tôi cam kết với các phần chia sẻ bằng cách sử dụng trình tạo được chọn độc lập g của $\mathbb{G}_q$ bằng cách xuất bản $v_i = g^{s_i}$. Hơn nữa, người phân phối xuất bản các ZKP không tương tác rằng các bản mã hoá $\hat{s}_i$ chứa cùng một phần chia sẻ như các cam kết v_i (sử dụng DLEQ như được mô tả trong Phần 2.6). Vì g được chọn độc lập với h, nên việc biết g^s không giúp kẻ tấn công truy xuất bí mật h^s trừ khi nó có thể tính toán log $log_h g$. Trong giai đoạn Xác minh, bất kỳ ai quan sát các phần chia sẻ được mã hóa công khai $\hat{s}_i$, các cam kết chia sẻ v_i và NIZK đều có thể kiểm tra xem phần chia sẻ được mã hóa có được tạo chính xác hay không bằng cách trước tiên xác minh rằng các phần chia sẻ trong $\hat{s}_i$ thực sự giống nhau trong v_i và sau đó thực hiện kiểm tra lý thuyết thông tin chỉ yêu cầu n phép tính. Việc kiểm tra này bao gồm việc chọn một Codeword $\boldsymbol{c}^{\perp} = (\boldsymbol{c}_1^{\perp}, \ldots, \boldsymbol{c}_n^{\perp})$từ mã kép $C^{\perp}$ tương ứng với mã

Reed Solomon C mà phương pháp Chia sẻ bí mật Shamir được sử dụng trong Phân phối là tương đương, tính tích bên trong của $c^{\perp}$ với Vectơ chia sẻ $(s_1,\ldots,s_n)$ bằng máy tính.

Giao thức π_{DDH}

Đặt g và h là hai trình tạo được chọn độc lập của một nhóm $\mathbb{G}_q$ có thứ tự q. Cho $H(\cdot)$ là một Oracle ngẫu nhiên. Đặt C là mã sửa lỗi tuyến tính tương ứng với lược đồ Chia sẻ Bí mật Shamir ngưỡng (n, t) và đặt $C^{\perp}$ là mã kép của nó.

Giao thức π_{DDH} được chạy giữa n bên $P_1,\ldots,P_n$, người phân phối D và trình xác minh bên ngoài V (thực tế là bất kỳ số lượng trình xác minh bên ngoài nào) có quyền truy cập vào sổ cái công khai nơi họ có thể đăng thông tin để xác minh sau này. Giao thức tiến hành như sau:

1. **Thiết lập:** Bên P_i tạo khóa bí mật $sk_i \leftarrow \mathbb{Z}_q$, khóa công khai $pk_i = h^{sk_i}$ và đăng ký khóa công khai pk_i bằng cách đăng nó lên sổ cái công khai, với $1 \le i \le n$.
2. **Phân phối:** Người phân phối D mẫu đầu tiên $s \leftarrow \mathbb{Z}_q$. Bí mật được xác định là $S = h^s$.D chọn $t-1$ hệ số $c_1,\ldots,c_{t-1} \leftarrow \mathbb{Z}_q$. D xây dựng một đa thức $p(x) = s + c_1x + c_2x^2 + \cdots + c_{t-1}x^{t-1}$ và tính các phần $s_i = p(i)$ cho $1 \le i \le n$. D mã hóa các chia sẻ dưới dạng $\hat{s}_i = pk_i^{s_i}$, tính toán các cam kết $v_i = g^{s_i}$ và tính toán $DLEQ\ (g, v_i, pk_i, s_i)$, với $1 \le i \le n$ từ một thử thách duy nhất e như được mô tả trong Phần 2.6, thu được $(e, z_1,\ldots, z_n)$. D công bố trên sổ cái công khai các phần chia sẻ được mã hóa $(s_1,\ldots,s_n)$ cùng với $PROOF_D = (v_1,\ldots,v_n, e, z_1,\ldots,z_n)$.
3. **Xác minh:** Đầu tiên, trình xác minh V kiểm tra xem $DLEQ\ (g, v_i, pk_i, \hat{s}_i)$ do D cung cấp có hợp lệ như được mô tả trong Phần 2.6 hay không. Nếu bằng chứng hợp lệ, V lấy mẫu một Codeword ngẫu nhiên $\boldsymbol{c}^{\perp} = (\boldsymbol{c_1}^{\perp},\ldots, \boldsymbol{c_2}^{\perp})$ của mã kép $C^{\perp}$ tương ứng với trường hợp chia sẻ bí mật Shamir ngưỡng (n, t) được D sử dụng và coi các phần chia sẻ là hợp lệ khi và chỉ khi biểu thức sau là đúng:
$$\prod_{i=1}^{n} v_i^{\boldsymbol{c}_i^{\perp}} = 1.$$
4. **Tái tạo:** Nếu một tập hợp gồm t hoặc nhiều bên Q muốn tái tạo lại bí mật, thì mỗi bên $P_i \in$ Q bắt đầu bằng cách xuất bản trong sổ cái công khai bản giải mã phần chia sẻ $\hat{s}_i = \hat{s}_i^{1/sk_i} = h^{s_i}$ và $PROOF_i = DLEQ\ (h, pk_i, \tilde{s}_i, \hat{s}_i)$ (cho thấy rằng phần chia sẻ được giải mã $\tilde{s}_i$ tương ứng với $\hat{s}_i$). Sau khi mọi bên trong Q xuất bản các phần chia sẻ được giải mã của họ và $PROOF_i$, trước tiên họ xác minh rằng các bằng chứng là hợp lệ và nếu kiểm tra này thành công, hãy tái tạo bí mật $S = h^s$ bằng phép nội suy Lagrange
$$\prod_{P_i \in \mathcal{Q}} (\tilde{s}_i)^{\lambda_i} = \prod_{P_i \in \mathcal{Q}} h^{p(i)\lambda_i} = h^{p(0)} = h^s,$$
trong đó $\lambda_i = \prod_{j \ne i} \frac{j}{j-i}$ là các hệ số Lagrange.

Hình 1. Giao thức π_{DDH}

$\prod_i v_i^{\boldsymbol{c}_i^{\perp}} = g^{\sum_i s_i \boldsymbol{c}_i^{\perp}}$ và kiểm tra kết quả là $g^0 = 1$. Giai đoạn Tái tạo tiến hành như trong [Sch99], với mỗi bên P_i "giải mã" phần chia sẻ của mình để lấy h^{s_i}, nó đã công bố cùng với bằng chứng rằng nó tương ứng với phần chia sẻ được mã hóa $\hat{s}_i$. Khi có sẵn các phần

chia sẻ được giải mã t, các bên có thể kiểm tra xem chúng có hợp lệ hay không và sử dụng phép nội suy Lagrange để tái tạo bí mật h^s. Giao thức được mô tả trong Hình 1.

3.1 Phân tích bảo mật

Lưu ý rằng các giai đoạn Thiết lập và Tái tạo hoàn toàn giống với các giai đoạn của [Sch99], trong khi giao thức của chúng tôi khác nhau trong các giai đoạn Phân phối và Xác minh, chúng tôi áp dụng kỹ thuật mới. Quan sát quan trọng là phần chia sẻ được mã hóa được tạo ra một cách độc hại $\hat{s}_1,...,\hat{s}_n$ sẽ chỉ vượt qua quy trình xác minh với xác suất $1/q$ cộng với lỗi xác thực của bằng chứng DLEQ, trong khi $v_1,...,v_n$ không tiết lộ thông tin nào về bí mật h^s theo giả định DDH (bằng một lập luận tương tự như [Sch99]).

Chúng tôi chính thức hóa những quan sát dưới đây. Đầu tiên chúng tôi xem xét Bảo mật IND1. Mặc dù chúng tôi sử dụng khái niệm Bảo mật IND1 thoải mái hoặc ứng dụng Beacon ngẫu nhiên của chúng tôi (trong đó không có bí mật nào được bảo toàn sau khi tái tạo), Giao thức π_{DDH} đạt được khái niệm Bảo mật IND1 mạnh mẽ hơn ban đầu của [RV05, HV09] (trong đó bảo mật đối với các bên bên ngoài tập hợp đủ điều kiện được đảm bảo ngay cả sau khi tái tạo) nếu việc xây dựng lại được thực hiện thông qua các kênh riêng tư giữa các bên trong tập hợp đủ điều kiện.

Định lý 1. *Theo giả định Decisional Diffie-Hellman, giao thức π_{DDH} là Bảo mật IND1 đối với một kẻ tấn công PPT tĩnh.*

Chứng minh. Chúng tôi chỉ ra rằng, nếu tồn tại một kẻ tấn công A_{Priv} có thể phá vỡ thuộc tính Bảo mật IND1 của giao thức π_{DDH}, thì tồn tại một kẻ tấn công A_{DDH} có thể sử dụng A_{Priv} để phá vỡ giả định Decisional Diffie-Hellman (DDH) với lợi thế tương tự. Không làm mất tính tổng quát, chúng tôi giả sử A_{Priv} làm hỏng t - 1 bên đầu tiên.

Cho $(g, g^{\alpha}, g^{\beta}, g^{\gamma})$ là một ví dụ của bài toán DDH. Rõ ràng nếu $\alpha = 0$ hoặc $\beta = 0$ thì vấn đề không đáng kể, vì vậy chúng tôi giả sử các giá trị này khác không. Bây giờ A_{DDH}, sử dụng A_{Priv}, có thể mô phỏng quá trình IND1 như sau:

1. Người thách thức đặt $h = g^{\alpha}$ và chạy giai đoạn Thiết lập của π_{DDH}. Đối với $t \leq i \leq n$, A_{DDH} chọn các giá trị ngẫu nhiên thống nhất $u_i \leftarrow \mathbb{Z}_p$ (các giá trị này có thể được coi là xác định ngầm định sk_i là $sk_i = u_i/\alpha$) và gửi các giá trị $pk_i = g^{u_i}$ đến A_{Priv}.
2. Đối với $1 \leq i \leq t$ -1, A_{Priv} chọn các giá trị ngẫu nhiên thống nhất $sk_i \leftarrow \mathbb{Z}_q$ và đặt $pk_i = h^{sk_i}$ và gửi giá trị này cho người thách thức.
3. Đối với $1 \leq i \leq t - 1$, người thách thức chọn các giá trị ngẫu nhiên thống nhất $s_i \leftarrow \mathbb{G}_q$ và đặt $v_i = g^{s_i}$ và $\hat{s}_i = pk_i^{s_i}$.

   Với $t \leq i \leq n$, nó tạo ra các giá trị $v_i = g^{p(i)}$ trong đó $p(x)$ là đa thức duy nhất bậc nhiều nhất tại t được xác định bởi $p(0) = \beta$ và $p(i) = s_i$ với $i = 1,...,t-1$. Lưu ý rằng A_{DDH} không biết β, nhưng nó biết $g^{\beta} = g^{p(0)}$ và $g^{s_i} = g^{p(i)}$ với $1 \leq i \leq t - 1$, vì vậy nó có thể sử dụng phép nội suy Lagrange theo phép tính để tính toán đầy đủ v_i. Nó cũng tạo ra các giá trị $\hat{s}_i = v_i^{u_i}$. Lưu ý rằng khi đó $\hat{s}_i = g^{u_i \cdot p(i)} = pk_i^{p(i)}$. Từ tất cả các giá trị được

tính toán, người thách thức hiện tạo bằng chứng DLEQ giống như người phân phối thực hiện trong giao thức PVSS. Cuối cùng, nó gửi tất cả thông tin này cùng với giá trị g^{γ} (đóng vai trò là x_b trong quá trình IND) đến A_{Priv}.

4. A_{Priv} dự đoán b'.

Nếu $b' = 0$, A_{DDH} đoán rằng $\gamma = \alpha \cdot \beta$. Nếu $b' = 1$, A_{DDH} đoán rằng γ là phần tử ngẫu nhiên trong $\mathbb{Z}_p$.

Thông tin mà A_{Priv} nhận được ở bước 3 được phân phối chính xác giống như việc chia sẻ giá trị $h^{\beta} = g^{\alpha \cdot \beta}$ với PVSS. Do đó, $\gamma = \alpha \cdot \beta$ khi và chỉ khi giá trị g^{γ} được gửi tới A_{Priv} là bí mật được chia sẻ bởi PVSS. Bây giờ dễ dàng nhận thấy rằng lợi thế dự đoán của A_{DDH} cũng giống như lợi thế của A_{Priv}.

Hai định lý sau đảm bảo tính chất có thể kiểm chứng của π_{DDH}.

Định lý 2. *Nếu người phân phối không xây dựng giá trị* (v_i, $\hat{s}_i$) *đúng trong giai đoạn Phân phối (nghĩa là* $\log_g v_i \neq \log_{pk_i} \hat{s}_i$ *đối với một vài i, hoặc* $\log_g v_i = \log_{pk_i} \hat{s}_i = s_i$ *với tất cả i nhưng các giá trị* s_i *không cấu thành chia sẻ hợp lệ bí mật nào đó trong* $\mathbb{Z}_q$ *với lược đồ chia sẻ bí mật Shamir ngưỡng* (*n*,*t*)*), thì điều này được phát hiện trong bước xác minh với xác suất ít nhất là 1-ε-1/q, với ε là lỗi về tính hợp lý của bằng chứng DLEQ.*

Chứng minh. Nếu xác minh DLEQ thành công, thì chúng ta có điều đó, ngoại trừ xác suất ε, với mọi $1 \leq i \leq n$, tồn tại s_i với $v_i = g^{s_i}$ và $\hat{s}_i = pk_i^{s_i}$. Bây giờ các giá trị s_i là phần chia sẻ hợp lệ với lược đồ chia sẻ bí mật Shamir ngưỡng (n, t) khi và chỉ khi Vectơ $\boldsymbol{v} = (s_1,...,s_n) \in C$. Giả sử $(s_1,...,s_n) \notin C$. Khi đó theo Bổ đề 1, vì $c^{\perp}$ được lấy mẫu ngẫu nhiên đều nên $\langle \boldsymbol{v}, \boldsymbol{c}^{\perp} \rangle \neq 0$ ngoại trừ với xác suất $1/q$. Nhưng sau đó

$$\prod_{i=1}^{n} v_i^{\boldsymbol{c}_i} = \prod_{i=1}^{n} g^{s_i \cdot \boldsymbol{c}_i^{\perp}} = g^{\langle \boldsymbol{v}, \boldsymbol{c}^{\perp} \rangle} \neq 1.$$

Do đó, nếu các giá trị s_i không phải là chia sẻ Shamir hợp lệ, thì kiểm tra không thành công với xác suất $1 - 1/q$.

Định lý 3. *Nếu một bên trong* Q *truyền thông tin phần chia sẻ được giải mã* $\tilde{s}_i$ *sai trong giai đoạn Tái tạo thì điều này được phát hiện bởi trình xác minh với xác suất 1-ε, trong đó ε là lỗi về tính hợp lý của bằng chứng DLEQ.*

Chứng minh. Điều này đơn giản theo định nghĩa vì một kẻ tấn công thành công trong việc cung cấp bằng chứng DLEQ cho một phần chia sẻ được giải mã không hợp lệ sẽ phá vỡ thuộc tính tình hợp lý của DLEQ.

4 PVSS dựa trên các cặp trong mô hình đơn giản

Trong phần này, chúng tôi xây dựng sơ đồ PVSS dựa trên giả định DBS trong mô hình đơn giản (không yêu cầu các Oracle ngẫu nhiên). Lược đồ này sử dụng các kỹ thuật của [HV09] để loại bỏ nhu cầu về các NIZK dựa trên Oracle ngẫu nhiên và thay vào đó sử dụng các cặp để kiểm tra xem các phần chia sẻ được mã hóa $\hat{s}_i$ có tương ứng với các phần chia sẻ đã cam kết v_i hay không và sau đó, kiểm tra xem các phần chia sẻ được giải

mã có tương ứng với $\hat{s}_i$ không. Chúng tôi sử dụng quy trình xác minh lý thuyết thông tin giống như trong sơ đồ dựa trên DDH. Giao thức được mô tả trong Hình 2.

Giao thức π_{DBS}

Đặt $\Lambda := (q, \mathbb{G}, \mathbb{G}_T, e)$ là một mô tả của một nhóm song tuyến tính và g, h là hai trình tạo được chọn độc lập của $\mathbb{G}$. Gọi C là mã sửa lỗi tuyến tính tương đương với sơ đồ chia sẻ bí mật Shamir ngưỡng (n, t) và $C^{\perp}$ là mã kép của nó. Giao thức π_{DBS} được chạy giữa n bên $P_1,...,P_n$, người phân phối D và trình xác minh bên ngoài V (thực tế là bất kỳ số lượng trình xác minh bên ngoài nào) có quyền truy cập vào sổ cái công khai nơi họ có thể đăng thông tin để xác minh sau này. Giao thức tiến hành như sau:

1. **Thiết lập:** Bên P_i tạo khóa bí mật $sk_i \leftarrow \mathbb{Z}_q$, khóa công khai $pk_i = h^{sk_i}$ và đăng ký khóa công khai pk_i bằng cách đăng nó lên sổ cái công khai, với $1 \le i \le n$.
2. **Phân phối:** Người phân phối D mẫu đầu tiên $s \leftarrow \mathbb{Z}_q$. Bí mật được xác định là $S = e(h,h)^s$. D chọn $t-1$ hệ số $c_1,...,c_{t-1} \leftarrow \mathbb{Z}_q$. D xây dựng một đa thức $p(x) = s + c_1x + c_2x^2 + \cdots + c_{t-1}x^{t-1}$ và tính các phần chia sẻ $s_i = p(i)$ với $1 \le i \le n$. D mã hóa các chia sẻ dưới dạng $\hat{s}_i = pk_i^{s_i}$ và tính toán các cam kết $v_i = g^{s_i}$, với $1 \le i \le n$. D đăng lên sổ cái công khai các phần chia sẻ được mã hóa $\hat{s}_1,...,\hat{s}_n$ và $PROOF_D = (v_1,...,v_n)$.
3. **Xác minh:** Trình xác minh V trước tiên kiểm tra xem $e(\hat{s}_i, g) = e(pk_i,v_i)$, với $1 \le i \le n$. Nếu việc kiểm tra này thành công, V lấy mẫu một Codeword ngẫu nhiên $\boldsymbol{c}^{\perp} = (\boldsymbol{c}_1{}^{\perp},\ldots,\boldsymbol{c}_n{}^{\perp})$ của mã kép $C^{\perp}$ tương ứng với trường hợp chia sẻ bí mật ngưỡng (n, t) của Shamir được D sử dụng và xem xét các chia sẻ hợp lệ khi và chỉ khi biểu thức sau là đúng:
$$\prod_{i=1}^{n} v_i^{c_i^{\perp}} = 1.$$
4. **Tái tạo:** Nếu một tập hợp gồm t hoặc nhiều bên Q muốn tái tạo bí mật, mỗi bên $P_i \in$ Q bắt đầu xuất bản trong sổ cái công khai phần mã chia sẻ được giải mãi của nó $\tilde{s}_i = \hat{s}_i{}^{1/sk_i} = h^{s_i}$ (ở đây $PROOF_i$ là một chuỗi rỗng). Một khi mọi bên trong Q xuất bản các phần chia sẻ được giải mã, trước tiên họ xác minh rằng $e(pk_i, \tilde{s}_i) = e(\hat{s}_i,h)$ với mọi $P_i \in$ Q. Nếu kiểm tra này thành công, họ tái tạo lại giá trị h^s bằng phép nội suy Lagrange:
$$\prod_{P_i \in Q} (\tilde{s}_i)^{\lambda_i} = \prod_{P_i \in Q} h^{p(i)\lambda_i} = h^{p(0)} = h^s,$$
trong đó $\lambda_i = \prod_{j \ne i} \frac{j}{j-i}$ là các hệ số Lagrange. Bí mật sau đó được tính là $S = e(h^s, h)$.

Hình.2. Giao thức π_{DBS}

4.1 Phân tích bảo mật

Như trong giao thức dựa trên DDH, lưu ý rằng các giai đoạn Thiết lập và Tái tạo hoàn toàn giống với các giai đoạn của [HV09], trong khi giao thức của chúng tôi khác nhau trong các giai đoạn Phân phối và Xác minh. Các phần chia sẻ được mã hóa được tạo ra một cách ác ý $\hat{s}_1,...,\hat{s}_n$ sẽ chỉ vượt qua quy trình xác minh với xác suất $1/q$, trong khi $v_1,...,v_n$ lại không tiết lộ thông tin nào về bí mật $e(h,h)^s$ nhưng lần này theo giả thiết BDS. Một lần nữa, chúng tôi nhận xét rằng Giao thức π_{DBS} đạt được khái niệm Bảo mật IND1 mạnh hơn ban đầu của [RV05, HV09] (trong đó bí mật đối với các bên bên ngoài tập

đủ điều kiện được đảm bảo ngay cả sau khi tái tạo) nếu việc tái tạo được thực hiện thông qua các kênh riêng tư giữa các bên trong tập đủ điều kiện.

Định lý 4. *Theo giả định DBS, giao thức π_{DBS} là Bảo mật IND1 với kẻ tấn công PPT tĩnh.*

Chứng minh. Tương tự như chứng minh của Định lý 1. Chúng tôi muốn chỉ ra rằng, nếu tồn tại kẻ tấn công A_{Priv} có thể phá vỡ thuộc tính riêng tư của giao thức π_{DBS} thì tồn tại kẻ tấn công A_{BDS} phá vỡ giả định Decisional Diffie-Hellman với lợi thế như nhau.

Giả sử chúng ta được cho một yêu cầu $(g, g^{\alpha}, g^{\beta}, T)$ và nhiệm vụ của A_{BDS} là đoán xem $T = e(g^{\alpha},g^{\alpha})^{\beta}$ hay $T = e(g^{\alpha},g^{\alpha})^{\gamma}$ với ngẫu nhiên $\gamma \leftarrow \mathbb{Z}_q$. A_{BDS} bây giờ mô phỏng quá trình IND1 cho π_{DBS} giữa một người thách thức A_{Priv} theo đúng các bước giống như A_{DDH} đã làm cho giao thức π_{DDH} trong chứng minh của Định lý 1, ngoại trừ bí mật bây giờ là $e(h,h)^{\beta} = e(g^{\alpha},g^{\alpha})^{\beta}$ và ở cuối bước 3. Người thách thức gửi giá trị T (chứ không phải g^{γ}).

Bây giờ nếu dự đoán của A_{Priv} là $b' = 0$ thì A_{BDS} đoán rằng $T = e(g^{\alpha},g^{\alpha})^{\beta}$. Nếu $b' = 1$, nó đoán rằng $T = e(g^{\alpha},g^{\alpha})^{\gamma}$ hoặc một $\gamma \leftarrow \mathbb{Z}_q$ ngẫu nhiên.

Dễ dàng nhận thấy rằng ưu điểm của A_{BDS} cũng giống như ưu điểm của A_{Priv}.

Định lý 5. *Nếu người phân phối không xây dựng các giá trị $(v_i, \hat{s}_i)$ ở dạng đúng trong giai đoạn Phân phối (nghĩa là $\log_g v_i \neq \log_{pk_i} \hat{s}_i$ đối với một vài i, hoặc $\log_g v_i = \log_{pk_i} \hat{s}_i = s_i$ với tất cả $1 \leq i \leq n$ nhưng các giá trị s_i không tạo thành phần chia sẻ hợp lệ một bí mật nào đó trong $\mathbb{Z}_q$ với sơ đồ chia sẻ bí mật Shamir ngưỡng (n, t), thì điều này được phát hiện trong bước xác minh với xác suất ít nhất là $1-1/q$.*

Chứng minh. Sự khác biệt duy nhất giữa chứng minh này và chứng minh cho giao thức π_{DDH} là ở đây chúng tôi không sử dụng chứng minh DLEQ để đảm bảo rằng $\log_{g_i} v_i = \log_{pk_i} \hat{s}_i$ với mọi i. Thay vào đó, điều này được xác minh bằng cách kiểm tra xem $e(\hat{s}_i,g) = e(pk_i,v_i)$ với $1 \leq i \leq n$. Lưu ý rằng nếu $a = \log_g v_i \neq \log_{pk_i} \hat{s}_i = b$ với một vài i, thì $e(\hat{s}_i,g) = e(pk_i,g)^b \neq e(pk_i,g)^a = e(pk_i,v_i)$ và phép kiểm tra thất bại với xác suất 1. Phần còn lại của chứng minh hoạt động chính xác như trong trường hợp của π_{DDH}.

Định lý 6. *Nếu một bên trong* Q *truyền đạt một phần chia sẻ được giải mã $\tilde{s}_i$ sai trong giai đoạn Tái tạo thì điều này được phát hiện bởi trình xác minh với xác suất 1.*

Chứng minh. Nếu $\tilde{s}_i = h^a$ với $a \neq s_i$ thì $e(pk_i,\tilde{s}_i) = e(pk_i,h)^a \neq e(pk_i,h)^{s_i} = e(\hat{s}_i, h)$.

5 Xây dựng Beacon ngẫu nhiên SCRAPE

Các lược đồ chia sẻ bí mật có thể kiểm chứng công khai có vô số ứng dụng như đã thảo luận trong [Sch99], trong số đó có các cuộc bầu cử có thể kiểm chứng trên toàn cầu, các phiên bản ngưỡng của mã hóa El Gamal và ký quỹ khóa phần mềm ngưỡng. Tuy nhiên, chúng tôi đặc biệt quan tâm đến việc xây dựng SCRAPE, một giao thức triển khai Beacon ngẫu nhiên phân tán được đảm bảo an toàn với đa số trung thực, sơ đồ PVSS và sổ cái công khai. SCRAPE về cơ bản là một giao thức tung đồng xu với phân

Giao thức π_{SCRAPE}

Giao thức π_{DDH} được chạy giữa n bên $P_1,...,P_n$ có quyền truy cập vào sổ cái công khai, nơi họ có thể đăng thông tin để xác minh sau này. Giao thức PVSS được sử dụng làm giao thức phụ và giả định rằng giai đoạn Thiết lập đã được thực hiện và khóa công khai pk_i của mỗi bên P_i đã được đăng ký trong sổ cái. Giao thức tiến hành như sau:

1. **Cam kết:** Đối với $1 \le j \le n$, bên P_j thực hiện giai đoạn Phân phối của giao thức phụ PVSS với tư cách là người phân phối có ngưỡng $t = \frac{n}{2}$, xuất bản các phần chia sẻ được mã hóa $\hat{s}_1{}^j,...,\hat{s}_n{}^j$ và thông tin xác minh $PROOF_D{}^j$ trên sổ cái công khai, đồng thời tìm hiểu bí mật ngẫu nhiên h^{s^j} và s^j. P_j cũng công bố cam kết về phép tính bí mật **Com** (s^j,r_j) (với độ ngẫu nhiên mới $r_j \leftarrow \mathbb{Z}_q$), với $1 \le j \le n$.
2. **Tiết lộ:** Đối với mọi tập hợp phần chia sẻ được mã hóa $\hat{s}_1{}^j,...,\hat{s}_n{}^j$ và thông tin xác minh $PROOF_D{}^j$ được xuất bản trong sổ cái công khai, tất cả các bên sẽ chạy giai đoạn Xác minh của giao thức phụ PVSS. Gọi C là tập hợp tất cả các bên đã công bố cam kết và phần chia sẻ hợp lệ. Khi các bên đã đăng các cam kết và phần chia sẻ hợp lệ của họ trên sổ cái, bên Pj sẽ mở cam kết của mình, đăng **Open** (s^j,r_j) trên sổ cái, với $j \in$ C.
3. **Khôi phục:** Đối với mọi bên $P_a \in$ C không xuất bản **Open** (s^a,r_a) trong giai đoạn Tiết lộ, bên P_j chạy giai đoạn Tái tạo của giao thức PVSS đăng $\tilde{s}^a{}_j$ và $PROOF_j{}^a$ vào sổ cái công khai với $1 \le j \le n$. Khi $\frac{n}{2}$ các phần chia sẻ được giải mã hợp lệ được xuất bản, mọi bên sẽ tái tạo lại h.

Độ ngẫu nhiên cuối cùng được tính là $\rho = \prod_{j \in C} h^{s^j}$.

Hình 3. Giao thức π_{SCRAPE}

phối đầu ra được đảm bảo (GOD - Guaranteed Output Delivery), nghĩa là kẻ tấn công không thể ngăn cản các bên trung thực nhận được đầu ra chính xác (ví dụ: bằng cách hủy bỏ trước khi thực hiện xong). Hơn nữa, SCRAPE có thể kiểm chứng công khai, nghĩa là bất kỳ ai cũng có thể phân tích các bản ghi giao thức trong quá khứ (và hiện tại) để xác minh rằng giao thức đang được thực thi chính xác. Lý do chúng tôi nhắm đến việc phân phối đầu ra được đảm bảo có hai mặt: 1. bảo vệ chống lại hành vi đặc biệt bất lợi và 2. chấp nhận các lỗi không phải lỗi của người dùng sau giai đoạn cam kết (ví dụ: mất điện). Khi được sử dụng để khởi động các giao thức đồng thuận dựa trên Blockchain như trong [KKR+16], π_{SCRAPE} phải chịu đựng những kẻ tấn công có thể gây ra sự mất đồng thuận tạm thời khiến người dùng kết thúc với các đầu ra ngẫu nhiên xung đột hoặc tạm thời "ngắt kết nối" người dùng khỏi mạng hoặc sổ cái công khai.

Chúng tôi tuân theo cách tiếp cận chung của [RBO89] để có được phân phối đầu ra được đảm bảo dựa trên chia sẻ bí mật có thể kiểm chứng, dựa trên các kế hoạch PVSS của chúng tôi để đạt được khả năng kiểm chứng công khai cho giao thức tung đồng xu cuối cùng. Cụ thể hơn, chúng tôi sử dụng các giao thức PVSS của mình để khởi tạo việc xây dựng [KKR+16], đề xuất kết hợp sơ đồ PVSS với sổ cái công khai để có thể tung đồng xu GOD có thể kiểm chứng công khai. Giao thức được mô tả trong Hình 3. Tính bảo mật của π_{SCRAPE} bắt nguồn từ tính bảo mật của cấu trúc chung của [KKR+16] và tính bảo mật của các giao thức của chúng tôi đã được chứng minh trong các phần trước. Chúng tôi giới thiệu người đọc đến [KKR+16] để thảo luận chi tiết về giao thức chung.

6 Độ phức tạp của thuật toán và các thí nghiệm

Trong phần này, chúng tôi thảo luận về hiệu quả cụ thể của các giao thức. Đầu tiên, chúng tôi trình bày độ phức tạp về tính toán và giao tiếp cụ thể của các sơ đồ, so sánh chúng với các giao thức của Schoenmakers [Sch99] và của Heidervand và Villar [HV09]. Tiếp theo, chúng tôi trình bày dữ liệu thử nghiệm từ việc triển khai nguyên mẫu của các giao thức được đề xuất, so sánh hiệu suất của các giao thức của chúng tôi so với Schoenmakers [Sch99].

	Phân phối	Xác minh		Tái tạo	
	Phép tính	Phép tính	Ghép nối	Phép tính	Ghép nối
[HV09]	$n + t$	nt	$2n$	t +1	$2t + 1$
Giao thức π_{DBS}	$2n$	n	$2n$	t +1	$2t + 1$
[Sch99]	$4n + t$	$nt + 4n$	-	$5t + 3$	-
Giao thức π_{DDH}	$4n$	$5n$	-	$5t + 3$	-

Bảng 1. Độ phức tạp tính toán cụ thể về số lượng phép tính và ghép nối cần thiết cho từng giai đoạn, xem xét n phần chia sẻ được tạo và t phần chia sẻ được sử dụng trong quá trình tái tạo.

Độ phức tạp tính toán: Trước tiên, chúng tôi bắt đầu bằng cách thảo luận về độ phức tạp tính toán của các giao thức PVSS, được so sánh với độ phức tạp tính toán của các giao thức [Sch99,HV09] về số lượng phép tính và ghép nối cần thiết cho từng giai đoạn trong Bảng 6. Lưu ý rằng phần chính cải tiến các giao thức của chúng tôi so với các nghiên cứu trước đó nằm ở giai đoạn xác minh, trong đó π_{DBS} yêu cầu ít hơn $n(t - 1)$ phép tính so với giao thức của [HV09] và π_{DDH} yêu cầu ít hơn nt phép tính giao thức của [Sch99], trong đó n là số lượng phần chia sẻ và t là ngưỡng.

Giai đoạn phân phối của π_{DBS} yêu cầu nhiều hơn $n - t$ phép tính so với giai đoạn phân phối của giao thức [HV09] và ít hơn $2n + t$ phép tính so với giao thức của [Sch99]. Giao thức π_{DDH} yêu cầu t ít phép tính hơn giao thức của [Sch99]. Số lượng phép tính nhỏ hơn theo yêu cầu của π_{DBS} có tác động mạnh mẽ trong việc cải thiện hiệu quả của giao thức này (như trong dữ liệu thử nghiệm). Các giai đoạn tái tạo của π_{DBS} (tương ứng với π_{DDH}) và giao thức của [HV09] (tương ứng với giao thức của [Sch99]) giống hệt nhau.

Lưu ý rằng đối với ứng dụng Beacon ngẫu nhiên, chúng tôi cần t = n/2, điều này chuyển thành chi phí bổ sung $n^2/2$ phép tính cần thiết cho giai đoạn xác minh của các giao thức trước đó. Các giao thức của chúng tôi loại bỏ chi phí bậc hai này, dẫn đến khả năng mở rộng tốt hơn nhiều. Ví dụ: nếu 10000 người dùng chạy SCRAPE dựa trên [Sch99], thì 50.004.000 phép tính được yêu cầu trong giai đoạn xác minh, trong khi việc khởi tạo SCRAPE của chúng tôi sẽ chỉ cần 50000 phép tính, đạt được mức tăng hiệu suất lý thuyết hơn 100 lần (mặc dù mức tăng hiệu suất thực tế là nhỏ hơn do chi phí hoạt động của các hoạt động khác như số học đa thức và I/O).

Độ phức tạp của giao tiếp: Trong Bảng 2, chúng tôi trình bày so sánh về độ phức tạp giao tiếp của sơ đồ của chúng tôi với các giao thức của [Sch99, HV09] về số lượng phần tử nhóm và phần tử vòng được yêu cầu cho từng giai đoạn. Sơ đồ dựa trên DDH của chúng tôi yêu cầu $2n$ phần tử nhóm và $n + 1$ phần tử vòng được đưa ra bởi người phân

phối trong khi cấu trúc dựa trên ghép nối của chúng tôi yêu cầu 2n phần tử nhóm nguồn. Trong các cấu trúc dựa trên DDH trước đây yêu cầu $n + t$ phần tử nhóm và $n + 1$ phần tử vòng, trong khi các cấu trúc dựa trên ghép nối trước đó yêu cầu $n + t$ phần tử nhóm nguồn. Chúng tôi lập luận rằng sự khác biệt này không đáng kể đối với ứng dụng Beacon ngẫu nhiên, vì $n = t/2$ trong kịch bản này, nghĩa là các kế hoạch của chúng tôi yêu cầu chi phí liên lạc bổ sung chỉ 0,5n phần tử nhóm.

	Phân phối		Tái tạo	
	$\mathbb{G}$	$\mathbb{Z}_p$	$\mathbb{G}$	$\mathbb{G}_p$
[HV09]	$n + t$	0	t	0
Giao thức π_{DBS}	2 n	0	t	0
[Sch99]	$n + t$	$n + 1$	t	t +1
Giao thức π_{DDH}	2n	$n + 1$	t	t +1

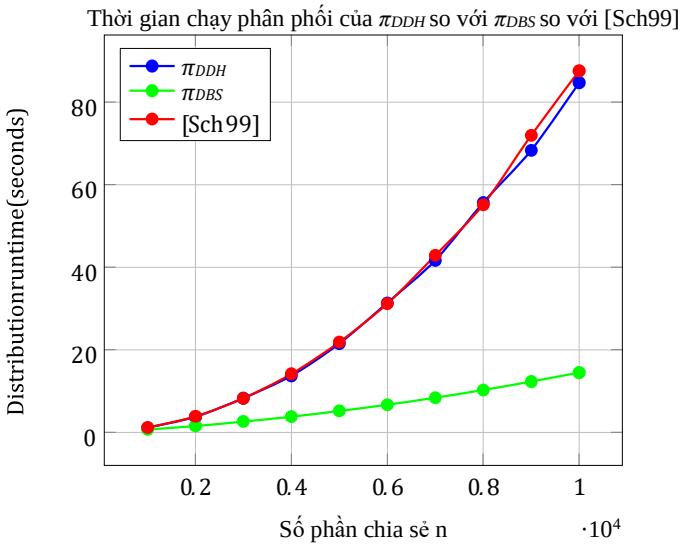
Bảng 2. Độ phức tạp giao tiếp cụ thể về số lượng phần tử của $\mathbb{G}$ và phần tử của $\mathbb{Z}_p$ cần thiết cho mỗi giai đoạn, xem xét n phần chia sẻ được tạo và t phần chia sẻ được sử dụng trong quá trình tái tạo (lưu ý rằng không cần giao tiếp cho giai đoạn tái tạo).

6.1 Các thí nghiệm

Để đánh giá hiệu suất cụ thể của các giao thức được đề xuất, chúng tôi đã tiến hành thử nghiệm với các triển khai Haskell nguyên mẫu [Han17, Ryb17] của π_{DDH}, π_{DBS} và sơ đồ PVSS của [Sch99]. Việc triển khai π_{DDH} và giao thức của [Sch99] dựa trên đường cong P256R1 trong khi việc triển khai π_{DBS} dựa trên thư viện mcl [Mit15], thư viện này triển khai ghép nối dựa trên đường cong Barreto-Naehrig 256 bit [BN06] Fp254BNb sử dụng các tham số và thuật toán được đề xuất trong [AKL+11].

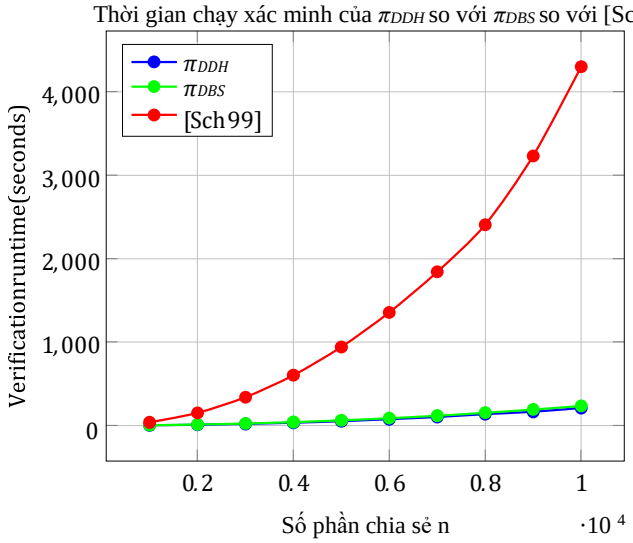
Lưu ý rằng một phiên bản π_{DBS} trên các ghép nối không đối xứng đã được triển khai để đạt được hiệu quả tốt hơn trong khi vẫn đảm bảo tính bảo mật. Có thể khởi tạo π_{DBS} trên một nhóm song tuyến tính không đối xứng $\Lambda := (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$ theo giả định co-DBS (tức là giả định rằng giả định DBS đúng trong cả hai nhóm nguồn). Điều này có thể được thực hiện bằng cách lấy mẫu $h \leftarrow \mathbb{G}_1$, $g, g' \leftarrow \mathbb{G}_2$ sao cho $pk_i = h^{sk_i} \in \mathbb{G}_1$ và $v_i \in \mathbb{G}_2$, thêm $pk'_i = g^{sk_i} \in \mathbb{G}_2$ vào giai đoạn thiết lập và xác định bí mật là $e(h^s, g')$. Giờ đây, việc kiểm tra tính hợp lệ của các phần chia sẻ được giải mã trong giai đoạn tái tạo có thể được thực hiện bằng cách sử dụng pk'_i thay vì pk_i bằng cách kiểm tra xem $e(\tilde{s}_i, pk'_i) = e(\hat{s}_i, g)$. Ngoài ra, việc kiểm tra này cũng có thể được thực hiện bằng cách sử dụng v_i mà không cần pk'_i bằng cách kiểm tra xem $e(\tilde{s}_i, g) = e(h, v_i)$, mặc dù yêu cầu mỗi bên phải tiết kiệm tất cả v_i cho đến giai đoạn tái tạo.

Chúng tôi phân tích thời gian thực hiện của từng giai đoạn của Giao thức π_{DDH}, Giao thức π_{DBS} và giao thức của [Sch99] khi xử lý n phần chia sẻ, cho n từ 1000 đến 10000. Chúng tôi đặt $t = \frac{n}{2}$, vì đó là ngưỡng được sử dụng trong ứng dụng Beacon ngẫu nhiên SCRAPE. Trong trường hợp của giai đoạn Tạo và Xác minh, chúng tôi phân tích thời gian thực hiện để tạo và xác minh n phần chia sẻ, trong khi đối với giai đoạn Tái tạo, chúng tôi phân tích thời gian thực hiện để giải mã và xác minh tính hợp lệ của t phần chia sẻ, sau đó sử dụng chúng để tái tạo bí mật. Thử nghiệm được chạy trên máy có CPU Intel(R) Core(TM) i7-7500U @ 2,70GHz và 16 GB RAM chạy 4.4.0-22-generic #40-Ubuntu SMP Linux Kernel.



Hình.4. Thời gian thực hiện các giai đoạn Phân phối của π_{DDH} so với π_{DBS} so với PVSS của Schoenmakers [Sch99] cho một số phần chia sẻ n từ 1000 đến 10000 và ngưỡng $t = \frac{n}{2}$.

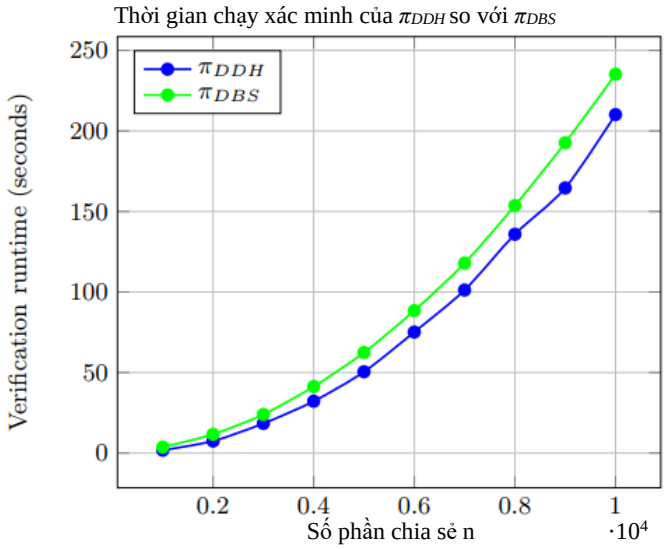
Giai đoạn phân phối: Mặc dù các giai đoạn phân phối của π_{DDH} , π_{DBS} và giao thức của [Sch99] rất giống nhau, nhưng π_{DDH} yêu cầu ít hơn t phép tính so với giao thức của [Sch99] và π_{DBS} yêu cầu ít hơn $2n + t$ phép tính so với giao thức giao thức của [Sch99]. Dữ liệu thử nghiệm của chúng tôi cho thấy rằng số lượng phép tính nhỏ được lưu bởi π_{DDH} không có tác động lớn đến hiệu suất cụ thể trong khi số lượng phép tính nhỏ hơn nhiều của π_{DBS} dẫn đến cải thiện hiệu quả rõ ràng. Thời gian thực hiện các giai đoạn phân phối của π_{DDH}, π_{DBS} và giao thức của [Sch99] được so sánh trong Hình 4.



Hình.5. Thời gian thực hiện các giai đoạn Xác minh của π_{DDH} so với π_{DBS} so với PVSS của Schoenmakers [Sch99] đối với một số cổ phần n từ 1000 đến 10000 và ngưỡng t = $n/2$.

Giai đoạn xác minh: Cải tiến chính của π_{DDH} và π_{DBS} so với giao thức của [Sch99] là giai đoạn xác minh giúp tiết kiệm số phép tính nt, tương đương với việc tiết kiệm $n^2/2$ số phép tính khi $t = \frac{n}{2}$, như trong các thử nghiệm của chúng tôi. Hình 5 so sánh thời gian thực hiện các giai đoạn xác minh của π_{DDH}, π_{DBS} và giao thức của [Sch99]. Lưu ý rằng trong trường hợp $n = 10000$ và $t = 5000$, lược đồ của chúng tôi nhanh hơn 20 lần so với lược đồ của [Sch99]. Trong Hình 5 cho thấy có vẻ như π_{DDH} và π_{DBS} có cùng thời gian thực hiện trong giai đoạn xác minh. Tuy nhiên, π_{DBS} có chi phí hoạt động so với π_{DDH} vì nó dựa trên các ghép nối. Thật thú vị, chi phí hoạt động này dưới 30% đối với $n \geq 3000$ và dưới 20% đối với $n > 5000$. Chúng tôi minh họa chi phí hoạt động của π_{DBS} so với π_{DDH} trong Hình 6, trong đó thời gian thực hiện giai đoạn xác minh của chỉ các giao thức của chúng tôi được so sánh.

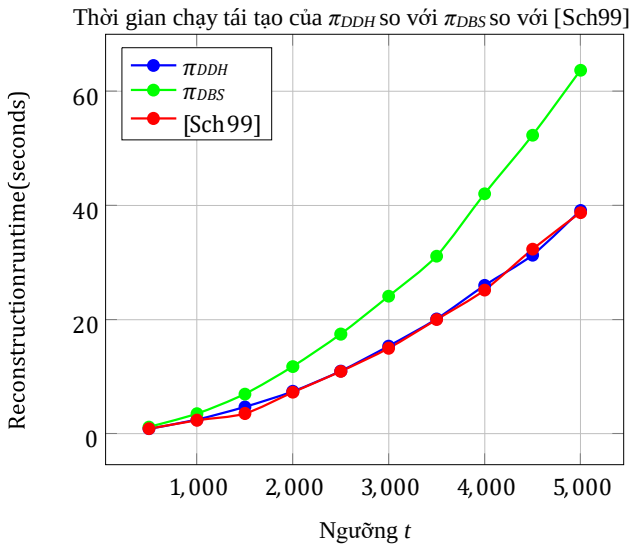
Giai đoạn tái tạo: Giai đoạn tái tạo của π_{DDH} và giao thức của [Sch99] hoàn toàn giống nhau, trong khi giai đoạn tái tạo của π_{DBS} yêu cầu $2n$ thao tác ghép nối so với tạo và kiểm tra n bằng chứng DLEQ. Trong các thử nghiệm này, chúng tôi xem xét thời gian thực hiện giải mã t phần chia sẻ, tạo và xác minh bằng chứng rằng t phần chia sẻ được giải mã này là hợp lệ và sử dụng chúng để nội suy bí mật cuối cùng. Chúng ta có thể thấy trong dữ liệu thử nghiệm rằng các hoạt động ghép nối có chi phí chung nhưng chi phí chung này nằm trong khoảng từ 40% đến 60% so với π_{DDH} và giao thức của [Sch99]. Dữ liệu thực nghiệm được so sánh trong Hình 7.



Hình.6. Thời gian thực hiện các giai đoạn Xác minh của π_{DDH} so với π_{DBS} cho một số phần chia sẻ n từ 1000 đến 10000 và ngưỡng $t = \frac{n}{2}$.

7 Phần kết luận

Chúng tôi đã giới thiệu lược đồ PVSS ngưỡng (*n*, *t*) đầu tiên trong đó chỉ yêu cầu các thao tác khóa công khai *O*(*n)* trong toàn bộ giao thức. Kỹ thuật chính của chúng tôi là giai đoạn xác minh lý thuyết thông tin mới và chúng tôi có thể sử dụng nó để xây dựng các sơ đồ trong ROM an toàn theo giả định DDH và trong mô hình đơn giản với các ghép nối theo giả định DBS. Sơ đồ PVSS hiệu quả này cho phép SCRAPE, một giao thức có thể mở rộng thực hiện một Beacon ngẫu nhiên với đa số trung thực. Yêu cầu *O*(*n*) hoạt động khóa công khai để xác minh vẫn chuyển thành *O*(n^2) hoạt động công khai cho mỗi bên trong SCRAPE (khi xác minh tất cả *n* phần chia sẻ từ tất cả *n* bên). Do đó, đây là một vấn đề mở thú vị để xây dựng một lược đồ PVSS tương tự chỉ yêu cầu một số lượng nhỏ các thao tác khóa công khai. Ngoài ra, kỹ thuật xác minh của chúng tôi yêu cầu người phân phối xuất bản ít nhất 2 phần tử nhóm *n* trên mỗi *n* phần chia sẻ, điều này có khả năng có thể giảm xuống *n* + *t* nhóm yếu tố như trong các nghiên cứu trước. Chúng tôi phân tích các kế hoạch của mình trong bối cảnh độc lập, để lại một công trình có thể kết hợp được như một công việc trong tương lai.



Hình.7. Thời gian thực hiện các giai đoạn Tái tạo của π_{DDH} so với π_{DBS} so với PVSS của Schoenmakers [Sch99] bao gồm quá trình giải mã, xác minh phần chia sẻ được giải mã và nội suy bằng cách sử dụng phần chia sẻ *t* làm đầu vào cho ngưỡng *t* từ 500 đến 5000.

Sự ghi nhận

Chúng tôi cảm ơn Vincent Hanquez vì đã triển khai π_{DDH} và giao thức của Schoenmakers [Sch99] và Andrzej Rybczak vì đã triển khai π_{DBS}. Chúng tôi cũng cảm ơn cả hai vì sự hỗ trợ của họ trong quá trình điểm chuẩn.

Tài liệu tham khảo

Adi08. Ben Adida. Helios: Bỏ phiếu kiểm toán mở dựa trên web. Trong Paul C. van Oorschot, biên tập viên, *Kỷ yếu của Hội nghị chuyên đề về bảo mật USENIX lần thứ 17, ngày 28 tháng 7 ngày 1 tháng 8 năm 2008, San Jose, CA, Hoa Kỳ*, trang 335–348. Hiệp hội USENIX, 2008.

AHO16. Masayuki Abe, Fumitaka Hoshino, và Miyako Ohkubo. Thiết kế kiểu I, chạy kiểu III: Chuyển đổi kiểu song tuyến tính nhanh và có thể mở rộng bằng cách sử dụng lập trình số nguyên. Trong Robshaw và Katz [RK16], trang 387–415.

AK+11. Diego F. Aranha, Koray Karabina, Patrick Longa, Catherine H. Gebotys và Julio Lo´pez. Các công thức rõ ràng nhanh hơn để tính toán các cặp trên các đường cong thông thường. Trong Kenneth G. Paterson, biên tập viên, *EUROCRYPT 2011*, tập 6632 của *LNCS*, trang 48–68. Springer, Heidelberg, tháng 5 năm 2011.

B+14. Vitalik Buterin và cộng sự. Một hợp đồng thông minh thế hệ tiếp theo và nền tảng ứng dụng phi tập trung. *sách trắng*, 2014.

BCG15. Joseph Bonneau, Jeremy Clark và Steven Goldfeder. Bitcoin như một nguồn ngẫu nhiên công khai. Cryptology ePrint Lưu trữ, Báo cáo 2015/1015, 2015. http://eprint.iacr.org/2015/1015.

BDF+15. Thomas Baignères, Cécile Delerablée, Matthieu Finiasz, Louis Goubin, Tancrède Lepoint, và Matthieu Rivain. Trap me if you can – đường cong triệu đô. Cryptology ePrint Lưu trữ, Báo cáo 2015/1249, 2015. http://eprint.iacr.org/2015/1249.

BDO14. Carsten Baum, Ivan Damgard, và Claudio Orlandi. Tính toán đa bên an toàn có thể kiểm toán công khai. Trong Michel Abdalla và Roberto De Prisco, người biên tập, *SCN 14*, tập 8642 của *LNCS*, trang 175–196. Springer, Heidelberg, tháng 9 năm 2014.

BGM16. Iddo Bentov, Ariel Gabizon, và Alex Mizrahi. Crypto không có bằng chứng công việc. Clark và cộng sự. [CMR+16], trang 142–157.

BLMR14. Iddo Bentov, Charles Lee, Alex Mizrahi và Meni Rosenfeld. Bằng chứng hoạt động: Mở rộng bằng chứng công việc của Bitcoin thông qua bằng chứng cổ phần [tóm tắt mở rộng]. *Đánh giá Đánh giá Hiệu suất SIGMETRICS*, 42(3):34–37, 2014.

BLN16. Daniel J Bernstein, Tanja Lange và Ruben Niederhagen. Dual ec: cửa sau tiêu chuẩn hóa. Trong *The New Codebreakers*, trang 256–281. Springer, 2016.

Blu81. Manuel Blum. Tung đồng xu qua điện thoại. Trong Allen Gersho, người biên tập, *CRYPTO'81*, tập Báo cáo ECE 82-04, trang 11–15. UC Santa Barbara, Bộ bầu cử. và Computer Eng., 1981.

BN06. Paulo SLM Barreto và Michael Naehrig. Các đường cong Elip thân thiện với cặp theo thứ tự nguyên tố. Trong Bart Preneel và Stafford Tavares, người biên tập, *SAC 2005*, tập 3897 của *LNCS*, trang 319–331. Springer, Heidelberg, tháng 8 năm 2006.

BR93. Mihir Bellare và Phillip Rogaway. Các phép Oracle ngẫu nhiên là thực tế: Một mô hình để thiết kế các giao thức hiệu quả. Trong V. Ashby, chủ biên, *ACM CCS 93*, trang 62–73. Nhà xuất bản ACM, tháng 11 năm 1993.

BT99. Fabrice Boudot và Jacques Traoré. Các kế hoạch chia sẻ bí mật có thể kiểm chứng công khai hiệu quả với khả năng khôi phục nhanh hoặc chậm. Trong Vijay Varadharajan và Yi Mu, người biên tập, *ICICS 99*, tập 1726 của *LNCS*, trang 87-102. Springer, Heidelberg, tháng 11 năm 1999.

CĐD+16. Ignacio Cascudo, Ivan Damgard, Bernardo David, Nico Dottling và Jesper Buus Nielsen. Tỷ lệ 1, thời gian tuyến tính và các cam kết UC đồng hình bổ sung. Trong Robshaw và Katz [RK16], trang 179–207.

CDE+16. Kyle Croman, Christian Decker, Ittay Eyal, Adem Efe Gencer, Ari Juels, Ahmed E. Kosba, Andrew Miller, Prateek Saxena, Elaine Shi, Emin Gu¨n Sirer, Dawn Song và Roger Wattenhofer. Mở rộng quy mô chuỗi khối phi tập trung - (Một báo cáo vị trí). Ở Clark và cộng sự. [CMR + 16], trang 106–125.

CGMA85. Benny Chor, Shafi Goldwasser, Silvio Micali và Baruch Awerbuch. Chia sẻ bí mật có thể kiểm chứng và đạt được tính đồng thời khi có lỗi (tóm tắt mở rộng). Trong *FOCS thứ 26*, trang 383–395. IEEE Computer Society Press, tháng 10 năm 1985.

Cle86. Richard Cleve. Các giới hạn về tính bảo mật của việc tung đồng xu khi một nửa số bộ xử lý bị lỗi (tóm tắt mở rộng). Trong Juris Hartmanis, biên tập viên, *Kỷ yếu của Hội nghị chuyên đề ACM thường niên lần thứ 18 về Lý thuyết máy tính, ngày 28-30 tháng 5 năm 1986, Berkeley, California, Hoa Kỳ*, trang 364–369. ACCM, 1986.

CMR+16. Jeremy Clark, Sarah Meiklejohn, Peter YA Ryan, Dan S. Wallach, Michael Brenner và Kurt Rohloff, biên tập viên. *Hội thảo FC 2016*, tập 9604 của *LNCS*. Springer, Heidelberg, tháng 2 năm 2016.

CP93. David Chaum và Torben P. Pedersen. Cơ sở dữ liệu ví với người quan sát. Trong Ernest F. Brickell, biên tập viên, *CRYPTO'92*, tập 740 của *LNCS*, trang 89–105. Springer, Heidelberg, tháng 8 năm 1993.

DMS04. Roger Dingledine, Nick Mathewson và Paul Syverson. Tor: Bộ định tuyến củ hành thế hệ thứ hai. Trong *Proceedings of the 13th Conference on USENIX Security Symposium - Volume 13*, SSYM'04, trang 21–21, Berkeley, CA, USA, 2004. Hiệp hội USENIX.

DPSW16. Jean Paul Degabriele, Kenneth G. Paterson, Jacob CN Schuldt và Joanne Woodage. Cửa sau trong trình tạo số giả ngẫu nhiên: Kết quả khả thi và không thể. Trong Matthew Robshaw và Jonathan Katz, biên tập viên, *CRYPTO 2016, Phần I*, tập 9814 của *LNCS*, trang 403–432. Springer, Heidelberg, tháng 8 năm 2016.

Fel87. Paul Feldmann. Một kế hoạch thực tế để chia sẻ bí mật có thể kiểm chứng không tương tác. Trong *FOCS thứ 28*, trang 427–437. IEEE Computer Society Press, tháng 10 năm 1987.

FO98. Eiichiro Fujisaki và Tatsuaki Okamoto. Một chương trình thực tế và an toàn có thể chứng minh được để chia sẻ bí mật có thể kiểm chứng công khai và các ứng dụng của nó. Trong Kaisa Nyberg, biên tập viên, *EUROCRYPT'98*, tập 1403 của *LNCS*, trang 32–46. Springer, Heidelberg, tháng 5/6 năm 1998.

FS87. Amos Fiat và Adi Shamir. Làm thế nào để chứng minh bản thân: Giải pháp thực tế cho các vấn đề nhận dạng và chữ ký. Trong Andrew M. Odlyzko, biên tập viên, *CRYPTO'86*, tập 263 của *LNCS*, trang 186–194. Springer, Heidelberg, tháng 8 năm 1987.

GKL15. Juan A. Garay, Aggelos Kiayias và Nikos Leonardos. Giao thức xương sống Bitcoin: Phân tích và ứng dụng. Trong Elisabeth Oswald và Marc Fischlin, biên tập viên, *EUROCRYPT 2015, Phần II*, tập 9057 của *LNCS*, trang 281–310. Springer, Heidelberg, tháng 4 năm 2015.

GPS06. SD Galbraith, KG Paterson và NP Smart. Ghép nối cho các nhà mật mã. Cryptology ePrint Archive, Report 2006/165, 2006. http://eprint.iacr.org/2006/165.

GRFJ14. Mainak Ghosh, Miles Richardson, Bryan Ford và Rob Jansen. Đường dẫn đến torcoin: các Altcoin bằng chứng về băng thông cho các rơle bù. Báo cáo kỹ thuật, Tài liệu DTIC, 2014.

Han17. Vincent Hanquez.pvss-haskell,2017. https://github.com/ input-output-hk/pvss-haskell.

HV09. Somayeh Heidarvand và Jorge L. Villar. Khả năng kiểm chứng công khai từ các cặp trong kế hoạch chia sẻ bí mật. Trong Roberto Maria Avanzi, Liam Keliher, và Francesco Sica, người biên tập, *SAC 2008*, tập 5381 của *LNCS*, trang 294–308. Springer, Heidelberg, tháng 8 năm 2009.

Jha11. Mahabir Prasad Jhanwar. Một chương trình chia sẻ bí mật thực tế (không tương tác) có thể kiểm chứng công khai. Trong Feng Bao và Jian Weng, biên tập viên, *Kinh nghiệm và Thực hành Bảo mật Thông tin - Hội nghị Quốc tế lần thứ 7, ISPEC 2011, Quảng Châu, Trung Quốc, ngày 30 tháng 5 - ngày 1 tháng 6 năm 2011. Kỷ yếu, tập 6672 của Bài* giảng *Khoa học Máy tính*, trang 273–287. Springer, 2011.

JVSN14. Mahabir Prasad Jhanwar, Ayineedi Venkateswarlu, và Reihaneh SafaviNaini. Chia sẻ bí mật có thể kiểm chứng công khai (không tương tác) dựa trên Paillier. *Designs, Codes and Cryptography*, 73(2):529–546, 2014.

KKR+16. Aggelos Kiayias, Ioannis Konstantinou, Alexander Russell, Bernardo David và Roman Oliynykov. Một giao thức Blockchain bằng chứng cổ phần an toàn đã được chứng minh. Cryptology ePrint Lưu trữ, Báo cáo 2016/889, 2016. http://eprint.iacr.org/2016/889.

KMS+16. Ahmed E. Kosba, Andrew Miller, Elaine Shi, Zikai Wen và Charalampos Papamanthou. Hawk: Mô hình Blockchain của mật mã và quyền riêng tư bảo vệ các hợp đồng thông minh. Trong *Hội nghị chuyên đề về bảo mật và quyền riêng tư của IEEE năm 2016*, trang 839–858. Nhà xuất bản Hiệp hội Máy tính IEEE, tháng 5 năm 2016.

LV08. Benoit Libert và Damien Vergnaud. Mã hóa lại Proxy bảo mật văn bản mật mã được chọn một chiều. Trong Ronald Cramer, người biên tập, *PKC 2008*, tập 4939 của *LNCS*, trang 360–379. Springer, Heidelberg, tháng 3 năm 2008.

LW15. Arjen K. Lenstra và Benjamin Wesolowski. Một sở thú ngẫu nhiên: con lười, kỳ lân và trx. Cryptology ePrint Archive, Báo cáo 2015/366, 2015. http://eprint.iacr.org/2015/366 .

Mau96. Ueli M. Maurer, chủ biên. *EUROCRYPT'96*, tập 1070 của *LNCS*. Springer, Heidelberg, tháng 5 năm 1996.

Mit15. Shigeo Mitsunari. mcl, 2015. https://github.com/herumi/mcl.

MS81. Robert J. McEliece và Dilip V. Sarwate. Chia sẻ bí mật và mã Reedsolomon. *Cộng đồng. ACM* , 24(9):583–584, 1981.

Nak08. Satoshi Nakamoto. Bitcoin: Một hệ thống tiền mặt điện tử ngang hàng. 2008.

nao91. Moni Naor. Cam kết giá trị của 1 bit sử dụng giả ngẫu nhiên. *Tạp chí Mật mã học*, 4(2):151–158, 1991.

Pai99. Pascal Paillier. Các hệ thống mật mã khóa công khai dựa trên các lớp độ dư hỗn hợp. Trong Jacques Stern, biên tập viên, *EUROCRYPT'99*, tập 1592 của *LNCS*, trang 223–238. Springer, Heidelberg, tháng 5 năm 1999.

PS96. David Pointcheval và Jacques Stern. Bằng chứng bảo mật cho các chương trình chữ ký. Trong Maurer [Mau96], trang 387–398.

Rab83. Michael O. Rabin. Bảo vệ giao dịch bằng đèn hiệu. *J. Máy tính. Hệ thống. Khoa học.*, 27(2):256–267, 1983.

RBO89. Tal Rabin và Michael Ben-Or. Chia sẻ bí mật có thể kiểm chứng và giao thức nhiều bên với đa số trung thực (tóm tắt mở rộng). Trong *ACM STOC thứ 21*, trang 73–85. Nhà xuất bản ACM, tháng 5 năm 1989.

RK16. Matthew Robshaw và Jonathan Katz, biên tập viên. *CRYPTO 2016, Phần III*, tập 9816 của *LNCS*. Springer, Heidelberg, tháng 8 năm 2016.

RV05. Alexandre Ruiz và Jorge Luis Villar. Chia sẻ bí mật có thể kiểm chứng công khai từ hệ thống mật mã của Paillier. Trong Christopher Wolf, Stefan Lucks, và PoWah Yau, biên tập viên, *WEWoRC 2005 - Hội thảo Tây Âu về Nghiên cứu Mật mã học, ngày 5-7 tháng 7 năm 2005, Leuven, Bỉ*, tập 74 của *LNI*, trang 98–108. GI, 2005.

Ryb17. Andrzej Rybczak. pvss-haskell, 2017. https://github.com/arybczak/pvss-haskell.

Sch99. Berry Schoenmakers. Một lược đồ chia sẻ bí mật có thể kiểm chứng công khai đơn giản và ứng dụng của nó đối với điện tử. Trong Michael J. Wiener, biên tập viên, *CRYPTO'99*, tập 1666 của *LNCS*, trang 148–164. Springer, Heidelberg, tháng 8 năm 1999.

Sha79. Adi Shamir. Làm thế nào để chia sẻ một bí mật. *Truyền thông của Hiệp hội Máy tính*, 22(11):612–613, tháng 11 năm 1979.

SJK+16. Ewa Syta, Philipp Jovanovic, Eleftherios Kokoris Kogias, Nicolas Gailly, Linus Gasser, Ismail Khoffi, Michael J. Fischer và Bryan Ford. Tính ngẫu nhiên phân tán chống sai lệch có thể mở rộng. Cryptology ePrint Archive, Report 2016/1067, 2016. http://eprint.iacr.org/2016/1067. Xuất hiện tại IEEE Security & Privacy 2017.

Sta96. Markus Stadler. Chia sẻ bí mật có thể kiểm chứng công khai. Trong Maurer [Mau96], trang 190–199.

SV15. Berry Schoenmakers và Meilof Veeningen. Tính toán đa bên có thể kiểm chứng toàn cầu từ các hệ thống mật mã đồng hình ngưỡng. Trong Tal Malkin, Vladimir Kolesnikov, Allison Bishop Lewko, và Michalis Polychronakis, biên tập viên, *ACNS 15*, tập 9092 của *LNCS*, trang 3–22. Springer, Heidelberg, tháng 6 năm 2015.

vdHLZZ15. Jelle van den Hooff, David Lazar, Matei Zaharia và Nickolai Zeldovich. Vuvuzela: Nhắn tin riêng tư có thể mở rộng chống phân tích lưu lượng. Trong *Kỷ yếu của Hội nghị chuyên đề lần thứ 25 về Nguyên tắc Hệ điều hành*, SOSP '15, trang 137–152, New York, NY, Hoa Kỳ, 2015. ACM.

WCGFJ12. David Isaac Wolinsky, Henry Corrigan-Gibbs, Bryan Ford và Aaron Johnson. Bất đồng về số lượng: Tạo quy mô ẩn danh mạnh mẽ. Trong *Kỷ yếu của Hội nghị USENIX lần thứ 10 về Thiết kế và Triển khai Hệ điều hành* , OSDI'12, trang 179–192, Berkeley, CA, Hoa Kỳ, 2012. Hiệp hội USENIX.

Người dịch: Nguyễn Văn Tú
Telegram: http://t.me/Tulibra
Link gốc: https://iohk.io/en/research/library/papers/scrape-scalable-randomness-attested-by-public-entities/